НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ННК «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Системного проектування
(повна назва кафедри)

«На правах рукопису»
УДК 004:004.453

«До захисту допущено»
Завідувач кафедри
_____ А.І.Петренко
(підпис) (ініціали, прізвище)
“___”_____2018 р.

Магістерська дисертація

зі спеціальності (спеціалізації) 122 – комп’ютерні науки та інформаційні технології (Системне проектування сервісів)
(код і назва спеціальності)

на тему: Реалізація швидкого парсеру повідомлень з різних гетерогенних джерел

Виконав: студент VI курсу, групи ДА-61м
(шифр групи)

Сергеєв Єгор Ігорович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник доцент, к.т.н. Кисельов Г.Д. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Розробка стартап-проекту доцент, к.т.н. Кисельов Г.Д. _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент проф., д.т.н. Аушева Н.М. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Інститут/факультет ННК "Інститут прикладного системного аналізу"
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти Другий (магістерський)
(перший (бакалаврський), другий (магістерський))

Спеціальність 122 – комп'ютерні науки та інформаційні технології
(Системне проектування сервісів)
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)
«___»_____ 2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Сергеєву Єгору Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема дисертації Розробка швидкого парсеру повідомлень з різних гетерогенних джерел

науковий керівник дисертації Кисельов Г.Д. , к.т.н., доц.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___» _____ 2018 р. №___

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження Гетерогенні джерела, великі масиви даних

4. Предмет дослідження Моделі, алгоритми та способи вирішення задачі обробки даних повідомлень з різних гетерогенних джерел

5. Перелік завдань, які потрібно розробити
   1. Провести дослідження соціальних мереж як гетерогенних джерел даних;
   2. Розглянути основні характеристики гетерогенних джерел даних;
   3. Провести аналіз алгоритмів для прогнозування даних;
   4. Провести аналіз моделі обробки даних гетерогенних джерел

даних;

5. Розробити програму-парсер повідомлень гетерогенних джерел з можливістю прогнозування майбутніх значень послідовності даних.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу презентація на тему “Обробка повідомлень з гетерогенних джерел”

7. Орієнтовний перелік публікацій
Сергеєв Є.І. Дослідження гетерогенних джерел даних на прикладі соціальних мереж та реалізація їх обробки і прогнозування даних. /Сергеєв Є.І. // Міжнародний науковий журнал “Інтернаука” 2018 – №8.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Реалізація стартап-проекту	Кисельов Г.Д. , к.т.н., доц.		

9. Дата видачі завдання 01.02.2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	01.02.2018	
2	Збір інформації	20.02.2018	
3	Аналіз вимог завдання, вибір методів і засобів розв’язання поставленої задачі	05.03.2018	
4	Дослідження характеристик гетерогенних джерел даних на прикладі різних соціальних мереж	10.03.2018	
5	Огляд методології обробки гетерогенних джерел даних	25.03.2018	
6	Розробка програми-парсеру з можливістю прогнозування послідовності даних	15.04.2018	
7	Оформлення дипломної роботи	22.04.2018	
8	Отримання допуску до захисту та подоча роботи в ДЕК	4.04.2018	

Студент _____ Сергеєв Є.І.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Кисельов Г.Д.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконану на тему: Розробка швидкого парсеру повідомлень з різних гетерогенних джерел

студентом: Сергеєвом Єгором Ігоровичем

Робота виконана на 99 сторінках, містить 17 іллюстрацій, 27 таблиць. При підготовці використовувалась література з 17 джерел.

Актуальність теми

На сьогоднішній день обсяг даних стає занадто великим для того, щоб була можлива їх обробка традиційними методами та алгоритмами. Таку проблему з великою кількістю даних ще називають великими даними (big data), а прикладом джерела такої кількості даних можуть бути гетерогенні джерела, такі як соціальні мережі.

Тому розробка програмного забезпечення, що може якісно і швидко завантажувати та обробляти дані з таких джерел є дуже актуальним напрямком дослідження, у час, коли кількість даних в гетерогенних джерелах зростає з небувалою швидкістю, а єдиного підходу для інтелектуального аналізу не існує.

Мета та задачі дослідження

Метою даної роботи є дослідження методів та технологій аналізу даних з гетерогенних джерел, з використанням різних засобів обробки великих даних. Результатом приведених досліджень є практична частина роботи, що становить собою реалізацію парсеру даних, що оброблює великі масиви даних та апробація з використанням сучасних програмних засобів.

Рішення поставленних задач та досягнуті результати

У даній роботі було запропоновано реалізацію парсера даних з таких соціальних мереж як LinkedIn, Jooble, Dou. Апробацію такого парсеру було виконано на локальному комп'ютері, а також в хмарному середовищі Google App Engine. З отриманих результатів можна зробити висновок, що розроблена програма-парсер вдало виконує свої функції, проте можливі її покращення, детально описані в роботі.

Об'єкт досліджень

Гетерогенні джерела, великі масиви даних

Предмет досліджень

Моделі, алгоритми та способи вирішення задачі обробки даних з гетерогенних джерел.

Методи досліджень

Для вирішення проблеми в даній роботі використовуються методи аналізу і синтезу, системного аналізу, порівняння, логічного узагальнення результатів, проектування логічних структур даних.

Наукова новизна

Наукова новизна роботи полягає у апробації сучасних моделей роботи з великими даними для вирішення задачі обробки даних з гетерогенних джерел, а саме соціальних мереж на прикладі LinkedIn, Dou та інших.

Практичне значення одержаних результатів

Отримані результати можуть використовуватись в майбутніх дослідженнях за напрямом створення моделей обробки даних з гетерогенних джерел, враховуючи переваги та недоліки, а також способи та методи продемонстровані в

даній роботі. Завдяки науковій публікації, яку розміщено в мережі інтернет, результати роботи будуть доступні за межами України.

Апробація рещзультатів дисертації

Результати досліджень опубліковано у міжнародному науковому журналі “Інтернаука”, випуск №8 2018 року.

Публікації

Сергеєв Є. І. Дослідження гетерогенних джерел даних на прикладі соціальних мереж та реалізація їх обробки і прогнозування даних. / Сергеєв Є.І. // Міжнародний науковий журнал “Інтернаука”, 2018 – №8.

Ключові слова

Інтелектуальний аналіз даних, гетерогенні джерела даних, розподілені обчислення, великі дані.

РЕФЕРАТ

НА МАГИСТЕРСКУЮ ДИСЕРТАЦИЮ

выполненную на тему: Построение быстрого парсера сообщений с разных гетерогенных источников

студентом: Сергеевым Егором Игоревичем

Работа выполнена на 99 страницах, содержит 17 иллюстраций, 27 таблиц. При подготовки использовалась литература из 17 источников.

Актуальность темы

На сегодняшний день количество даных становится слишком большим для того, чтобы была возможность их обработки традиционными методами и алгоритмами. Такую проблему с большим количеством даных еще называют большими данными (big data), а примером источника такого количества даных могут быть гетерогенные источники, такие как социальные сети.

Поэтому разработка програмного обеспечения, которое может качественно и быстро загружать и обрабатывать данные с таких источников является очень актуальным направлением исследований, во время, когда количество даных в гетерогенных источниках увеличивается с невообразимой скоростью, а единственного подхода для интеллектуального анализа не сувществует.

Цель и задачи исследования

Целью данной работы является исследование методов и технологий анализа даных с гетерогенных источников, с использованием различных способов обработки больших данных. Результатом приведенных исслеований является практическая часть работы, что являет собой реализацию парсера даных, который обрабатывает большие массивы соответсвующих данных и апробация с использованием современных програмных средств.

Решение поставленных задач и полученные результаты

В даной работе было предложено реализацию парсера данных с таких социальных сетей как LinkedIn, Jooble, Dou. Апробацию такого парсера было произведено на локальном компьютере, а также в облачной среде Google App Engine. С полученых результатов можна сделать следующие выводы, что разработанная програма-парсер справляется со своими задачами, однако её можно улучшить, соответствующие улучшения представленны в работе.

Объект исследования

Гетерогенные источники, большие массивы данных

Предмет исследования

Модели, алгоритмы и способы решения задачи обработки даных с гетерогенных источников.

Методы исследований

Для решения проблемы в работе используются методы анализа и синтеза, системного анализа, сравнения, логичного обобщения результатов, проектриования логических структур данных.

Научная новизна

Научная новизна работы содержит апробацию современных моделей работы с большими данными для решения задачи обработки данных с гетерогенных источников, а именно социальных сетей на примере Twitter и Facebook.

Практическое значение полученных результатов

Полученные результаты могут использоваться в последующих иследованнияв в направлении создания моделей обработки даных с гетерогенных источников, используя недостатки и приемущества, а также способы и методы

продемонстрированные в данной работе. С помощью научной публикации, которую выставлено в сети интернет, результаты работы будут доступны за пределами Украины.

Апробациия результатов дисертации

Результаты исследований опубликованы в международном научном журнале “Интернаука”, выпуск №8 2018 года.

Публикации

Сергеев Е.И. Исследование гетерогенных источников даных на примере социальных сетей и реализация их обработки и прогнозирования данных. / Сергеев Е.И. // Международный научный журнал “Интернаука”, 2018 – №8.

Ключевые слова

Интеллектуальный анализ данных, гетерогенные источники данных, распределенные вычисления, большие данные.

ABSTRACT

ON MASTER'S THESIS

on topic: Implementation of high-speed data parser from different heterogeneous data sources

student: Yehor I. Serheiev

Work carried out on 99 pages containing 17 figures, 27 tables. The paper was written with references to 17 different sources.

Topicality

Nowadays amount of data becomes too large to be processed by traditional methods and algorithms. Such problem with large amount of data is also known as a "big data", example of such data sources can be heterogeneous sources such as social networks.

Therefore the development of software that can efficiently and quickly download and process data from such sources is a relevant area of research, in time when the amount of data in heterogeneous sources is increasing at an unpredictable rate, and there is no single approach for intellectual analysis.

Purpose

The purpose of this work is to study the methods and technologies of data analysis from heterogeneous sources, using various methods for large data sources processing. Practical part of current work is the result of these research which is an implementation of data parser that processes large data sets and it's testing using modern software tools and solutions.

Solution

Implementation of high-speed data parser from social networks like LinkedIn, Jooble and Dou was proposed in this thesis. Testing of this parser was performed on

local environment as well as in the cloud environment such as Google App Engine. It can be concluded that the developed parser successfully performs it’s functions but requires improvements described in thesis.

The object of research

Heterogenous data sources, big data.

The subject of research

Models, algorirhms and soutions for solving tasks such as heterogenous data sources processing.

Research methods

To solve the problem in this research such methods were used: analysis and synthesis, system analysis, comparison, logical generalization of the results, design logical data structures.

Scientific novelty

The scientific novelty lies in approbation of modern models for heterogenous data source processing like LinkedIn and Dou social networks.

The practical value of research

The results obtained can be used in future research in the area of data processing from heterogeneous data sources models creation, taking into account advantages and disadvantages as well as methods demonstrated in this thesis. Thanks to a scientific publication posted on the Internet, the results from current thesis will be available outside of Ukraine.

Research results approbation

Research results presented in the international scientific journal “Internauka”, issue №8 – 2018.

Publications

Serheiev Y. Hererogeneous data sources like social media research and implementaion of processing and forecasting of social media data. / Yehor Serheiev. // International Scientific Journal “Internauka”, 2018 – №8.

Keywords

Intellectual data analysis, heterogeneous data sources, distributed computing, big data.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ARIMA – Auto-Regressive Integrated Moving

ES – Exponential smoothing

HDFS – Hadoop distributed file system

JSON – Javascript object notaion

JWT – JSON Web Token

SQL – Structured Query Language

ВСТУП

В сучасному світі люди в силу прогресу контактують один з одним, використовуючи Інтернет. Використовуючи такі засоби як соціальні мережі, люди отримали змогу спілкуватися з іншими з різних частин світу. Це призвело до того, що соціальні мережі стали простором формування та зікріплення різних скультурних стереотипів.

Наприкінці минулого століття значно змінилися способи та стиль життя людини, що відтоді має змогу налагоджувати необхідні для неї контаки, що відповідають її інтересам. Відтепер, щоб мати змогу взаємодіяти з представниками споріднених та інших культур, людині не потрібно виходити з дому, адже завдяки Інтернету кожен має доступ до безлічі ресурсів по всьому світу. Одним з таких способів взаємодії є соціальні мережі, що майже не піддаються зовнішньому контролю, не мають централізованності, а кожен в праві діяти як він вважає за потрібне. Сучасна людина не просто прагне дізнатися про певні події, що виникають навколо неї, вона хоче їх створювати [3].

Дані, доступні в соціальних мережах, можуть дати уявлення про людину, громади та суспільство в цілому, що раніше було неможливим в таких глобальних масштабах. Такі цифрові медіа дані виходять за межі фізичного світу для вивчення людських відносин і допомагають визначати популярні соціальні та політичні настрої для регіональних груп без використання опитувань. Соціальні мережі регулярно фіксують основні маркетингові тренди, настрої та напрями та є ідеальними джерелами для вивчення та обробки. Однак, отримання необхідної інформації з соціальних мереж є дуже важким через конкретні виклики, що несуть такі дані. Методи інтелектуального аналізу даних можуть допомогти ефективно оброблювати інформацію з соціальних мереж та обійти три основні проблеми таких даних.

По-перше, дані із соціальних мереж мають величезний розмів, для прикладу, сотні мільйонів користувачів такої мережі як Facebook кожного дня

створюють безліч даних. Без спеціального, програмного методу обробки та необхідних алгоритмів та способів, процедура обробки таких даних стає неймовірно ресурсозатратною і не виконається в розумні проміжки часу [20].

По-друге, соціальні мережі мають в собі "шум", спам, незмістовні дані, пусті повідомлення. Дані такого типу необхідно видаляти з набору чи ігнорувати, що є складною процедурою, для якої потрібно використовувати спеціальні алгоритми [20].

По-третє, дані із соціальних мереж мають динамічний характер, їм властиві часті оновлення даних, що створює безліч проблем щодо актуальності даних. Інші типи наборів даних не з соціальних мереж містять лише деякі із зазначених проблем, однак їм не властиві усі проблеми зразу. Для прикладу, масив веб-сторінок генерує дані, що є величезними за об'ємом та містять шуми, проте, порівняно з соціальними мережами, вони не мають динамічного характеру, а саме властивості постійно оновлюватись. Для приклажу, звичайне блогове повідомлення та відповідні коментарі, до того ж кожен коментар може містити різного типу дані, як наприклад, текст, відео, зображення [20].

Ще одним аспектом даних, що є у соціальних мережах – це їх реляційний характер, що ускладнює аналіз обробки даних, проте вони це не є новою проблемою для пошуку даних [20].

Методи інтелектуального аналізу викликані допомогти дослідникам та практикам обійти всі названі проблеми, використання таких технік та методів дозволяє покращити результати пошуків пошукових систем, допомогти в реалізації спеціалізованої маркетингової політики для бізнесу, надати нові знання про соціальні структури для соціологів, персоналізувати веб-системи для користувачів завдяки розвидку рекомендаційних систем і навіть допомогти розпізнавати спам та захищати від нього. Деякі з методів інтелектуального аналізу даних, як наприклад, прогностичний аналіз є описаними детально в даній дипломній роботі [20].

В першій частині дипломної роботи описано мету та наукову новизну та інтерес до досліджень соціальних мереж. Розглядаються основні напрями дослідження, а також проблеми досліджень викликані динамікою та об’ємом даних соціальних мереж.

В другій частині диплому описано методологію обробки природної мови та розглянуто основні методи та алгоритми вилучення необхідної інформації з даних соціальних мереж. Також буде розглянуто методи та алгоритми прогностичного аналізу для прогнозування трендів, та вибір архітектури для виконання обробки даних.

В третій частині описана реалізація програми, її основні можливості та ключові частинни, використані технології та результати роботи. Показано інтерфейс та процес отримання прогностичних даних по трендам фреймворків та спеціалізації з соціальних мереж пошуку кандидатів та роботи.

В четвертій частині описано стартап-проект, його характеристики, маркетингова програма, ринкові можливості та стратегії виходи на ринок.

1 СОЦІАЛЬНІ МЕРЕЖІ ЯК ПРИКЛАД ГЕТЕРОГЕННИХ ДЖЕРЕЛ ДАНИХ

1.1 Соціальні мережі як приклад гетерогенних джерел даних

Гетерогенність даних – характеристика даних, що показує несхожість та неоднорідність даних, тобто такі дані, які не мають спільних атрибутів чи властивостей, наприклад, взяті з різних джерел даних. В статистиці існує таке поняття як гетерогенна вибірка, тобто така, яка складається з неоднорідних об’єктів, як наприклад, вибірка з сільського та міського населення і т.д., такі вибірки мають різну ступінь гетерогенності. Дані з гетерогенних джерел є цікавими для досліджень, оскільки можуть формувати картину об’єкта в цілому, яку не можна отримати з одного джерела, що описує однорідні властивості об’єкта.

На противагу гетерогенності існує протилежне поняття – гомогенності, тобто властивості однорідності об’єктів. Гомогенні джерела складаються зі схожих за властивостями об’єктів, які можна обробити задля виконання статистичного аналізу, тощо.

Соціальна мережа – соціальна структура, що було створено організаціями або індивідами, яка відображає різноманітні зв’язки між учасниками через соціальні взаємовідносини, починаючи від випадкових знайомств, закінчуючи тісними родинними вузами[1].

Теорія соціальних мереж розглядає соціальні взаємовідносини в термінах вузлів та зв’язків. Вузли є уособленням акторів у мережі, а зв’язки відповідають стосункам між акторами, при чому може існувати багато типів зв’язків між вузлами. В своїй найпростішій формі, соціальна мережа відображеням всіх зв’язків, що мають відношення до дослідження між вузлами[2].

Хоча поняття соціальних мереж є досить широким, в контексті даної роботи, соціальною мережею будуть вважатися веб-сайти, так і мобільні інтернет-

програми, які дозволяють створювати, отримувати доступ і виконувати обмін користувальницьким контентом, який є повсюдно доступним. Крім соціальних онлайн мереж (як наприклад, Twitter і Facebook), для зручності, також термін “соціальна мережа” буде використовуватись і для опису RSS каналів, блогів, вікі та веб-сайтів новин, які, зазвичай, складаються з неструктурованого тексту, що публічно доступний через Інтернет.

До основних функцій соціальних мереж модна віднести такі – створення індивідуальних профілів користувачів (акторів), в яких буде міститися певна інформація про користувача, при чому різновид даних конкретно залежить від типу соціальної мережі, однією з основниз функцій є взаємодія користувачів, типовим засобом такої взаємодії є перегляд профілю користувача, написання коментарів чи перегляд внутрішньої пошти, наприклад, читачі блогу можуть активно обговорювати статтю автора, чи спростовувати інформацію в ній. Ще однією функцією є можливість досягнення спільної мети шляхом кооперації, тобто пошук людей зі схожими поглядами, пошук друзів та обмін ресурсами різного типу. Соціальні мережі є джерелом задоволення потреб ща рахунок накопичення ресурсів, таких як друзів чи інформації[1].

Соціальні мережі можна розрізняти за двома основними видами – публічні та закриті. Прикладом публічної соціальної мережі є звичайний блог, що викладено в Інтернет, в якому автор ділиться інформацією, яку не вважає приватною, чи такою, що не може зашкодити своїй приватності чи приватності іншої особи. В той же час приватною або закритою соціальною мережею можна назвати мережі доступ до яких має певне коло користувачів. Прикладом такої мережі є корпоративний портал, в якому співробітники компанії можуть переглядати профілі один одного, бронювати кімнати для мітингів, створювати запити на технічне обслуговування і т.д. В даній роботі будуть розглядатися лише публічні соціальні мережі, а також дані, що вважаються відкритими і публічно доступними.

Соціальні мережі можна також поділити за такими типами:

1. Інформаційні соціальні мережі – співтовариство людей, які шукають рішення повсякденних проблем. Це ключовий сегмент для маркетологів. Такі ресурси часто бувають пов’язані з конкретними підприємствами і компаніями, які зацікавлені в нових каналах просування своїх товарів та послуг[3].
2. Освітні соціальні мережі – поєднують студентів і орієнтовані на взаємодію з метою надання допомоги в реалізації академічних проектів, проведення наукових досліджень або взаємодії з викладацьким складом.
3. Дослідницькі соціальні мережі – орієнтовані на академічну аудиторію. Такі мережі дозволяють академічним дослідникам займатися спільними науковими дослідженнями в різних областях[3].
4. Соціальні мережі, що спеціалізуються на хобі – є неофіційними і мають змішані групи користувачів. Єдиною метою цих груп – обмін інформацією та враженнями за спільними інтересами[3].
5. Соціальні мережі для аматорів новин поєднують користувачів, що прагнуть генерувати контент у вигляді цікавих певним групам новин, коментарів, а також освітленню останніх подій у різних областях[3].
6. Соціальні мережі для аматорів кіно, музики спрямовані на об’єднання людей, які мають схожі інтереси в галузі мистецтва – кіно, музика, живопис, скульптура та інші. За рахунок спеціалізації вони мають набагато більше спільної інформації та інтересами, такі соціальні мережі надають можливість спілкування однодумців без участі сторонніх[3].
7. Соціальні мережі пошуку роботи та кандидатів – такі соціальні мережі дуже широко поширені і є чи не найефективнішим методом пошуку роботи та кандидатів для фірм та працівників, вони мають зручний інтерфейс подачі резюме на вакансію, а також засоби розміщення вакансій для огляду та пошуку. Дані соціальні мережі є ключовими і є основними джерелами даних для даної роботи[3].

Соціальні мережі особливо важливі для досліджень різних соціальних наук, що використовують кількісні методи (наприклад, обчислювальна статистика, машинне навчання) та так звані “big data” методи для обробки даних. Дані соціальних мереж характеризуються трьома основними параметрами, а саме – розмір, шум та динамізм. Об’ємність та динамічність наборів даних соціальних мереж вимагає спеціалізованої автоматичної обробки для аналізу даних протягом невеликого проміжку часу. Така характеристика даних соціальних мереж створює ряд дослідницьких проблем і піднімає низку питань, що стоять перед реалізацією методів вилучення даних, загалом для вирішення таких проблем використовуються такі методи аналізу[6]:

1. Структурний аналіз та аналіз зв’язків – це аналіз поведінки зв’язків в соціальних мережах, щоб з’ясувати відповіни вузли, зв’язки між вузлами і т.д.
2. Динамічний аналіз та статичний аналіз – статичний аналіз використовується в бібліографічних мережах ніж в потокових, в статичному аналізі припускається, дані соціальних мереж змінюються поступово з часом і аналіз цілої мережі можна провести в пакетному режимі. Однак для таких соціальних мереж, в яких лані генеруються з високою швидкістю і в великих об’ємах використовуэться динамічний аналіз.

Цікаво, що методи обробки даних також потребують великих об’ємів даних задля того, аби отримати необхідні шаблони з отриманих даних. З цього слідує, що соціальні мережі є ідеальними джерелами даних для обробки. Цей факт формує сприятливий фактор для розширеного пошуку в пошукових системах, а також допомагає кращому розумінню соціальних даних для досліджень та організаційних функцій.

1.2 Основні напрями дослідження соціальних мереж

Оскільки соціальні мережі включають в себе неймовірну кількість даних, що оновлюється з кожним днем, то таке джерело даних є ключовим в дослідженнях соціальних наук. Пошук та обробка даних з соціальних мереж є складною задачею, через безліч факторів, в тому числі неоднозначність людської мови, багатозначність слів, псевдоніми для одного і того самого користувача, неправильне зображення даних та двозначність взаємовідносин між користувачами. В даному розділі розглянуто основні напрями та методи дослідження соціальних мереж.

Теорія графів. Теорія графів, мабуть, була чи не основним методом дослідження в ранній історії дослідження соціальних мереж. Такий підхід застосовується для аналізу соціальних мереж з метою визначення важливих особливостей мережі – зв’язків та вузлів мережі. Користувачі соціальної мережі визначаються як основні чинники впливу на діяльність або думку інших користувачів способами, доступними в соціальній мережі. Використання теорії графів для аналізу соціальних мереж виявилося ефективним для великомасштабних наборів даних. В основному, можна виділити два основні напрями дослідження з використанням теорії графів[6]:

1. Виділення спільнот за допомогою ієрархічної кластеризації. Формація сіпльнот є чи не найважливішою характеристикою соціальних мереж. Користувачі зі спільними інтересами формують спільноти, і як результат відображають сильну секційну структуру. Виявлення спільнот як в соціальних мережах, так і в реальному світі є дуже складною процедурою, адже їх важко виявити. Застосування відповідних інструментів для виявлення та розуміння поведінки спільнот в мережі є важливим, адже це можна застосувати для моделювання динаміки доменів до яких належать користувачі. Загалом, можна використовувати різні методи кластеризації для пошуку спільнот, однак одною з найефективніших є метод ієрархічної

кластеризації, що являє собою поєднання багатьох методів для групування вузлів в мережі для виявлення силу груп та їх зв’язків, що згодом використовується для розділення мережі на спільноти.

2. Створення рекомендацій для спільнот соціальної мережі. Виходячи з взаємозв’язків в групах соціальних мереж, може бути використана техніка колаборативної фільтрації, яка формує одого з трьох класів рекомендаційних систем і може бути використана для асоціацій між користувачами. Елемент (товар, предмет і т.д.) може бути рекомендованим для користувачів на основі рейтингу його взаємного зв’язку. Основним недоліком колаборативної фільтраціх є ранджування даних, тому можна використовувати рекомендаційну систему на основі вмісту для дослідження структури даних і створення рекомендацій, однак зачасту, використовуються обидва підходи. Прикладом такого використання є так званий підхід EntreeC, що використовується для рекомендування ресторанів.

Аналіз поглядів користувачів соціальних мереж. За останніми данними, в середньому більше 1,2 мільйона постів з’являється кожного дня в популярних соціальних мережах. Щохвилини з’являється безліч даних з інформацією, думками, поглядами користувачів соціальних мереж на товари, послуги, особисті та глобальні питань. Показники думок користувачів часто є досить переконливими і можуть використовуватися як основний чинник прийняття рішення про вибір певних товарів чи послуг або навіть схвалення кандидата на виборах. Незважаючи на те, що погляди користувачів в Інтернеті можна виявити, використовуючи традиційні способи, така форма обробки даних буде давати неадекватні результати, враховуючи великий обсяг інформації, що створюється на сайтах соціальних мереж. Цей факт підкреслює важливість та актуальність розробки нових технік вилучення даних з соціальних мереж, розроблено безліч методів аналізу викладених думок, поглядів і т.д., включаючи використання сентиментного аналізу (аналіз тональності тексту), колекції простих методів

підрахунку, алгоритми машинного навчання. Розглянемо основні методи обробки поглядів користувачів[6]:

1. Аспектний або характеристичний аналіз – процес обробки області об’єкта, що був оглянутий користувачем, згодом оброблені аспекти даних сумуються та оброблюються для визначення так званої полярності всіх поглядів, для визначення чи є вони негативними або позитивними. Не всі погляди легко обробити, оскільки деякі з них можуть бути неоднозначними. В основному дослідження з використанням аспектного та характеристичного аналізу використовуються для блогів та форумів, оскільки на таких сайтах створюються дискусії різних типів. Наприклад, можна використовувати так звану статистику “палець вгору” та “палець вниз”, однак цього буває недостатньо і буває необхідно оброблювати весь текст для пошуку речень з поглядами та використовувати методи аспектного аналізу на них.
2. Методологія визначення поглядів та підведення підсумків (підсумовування). Об’єм інформації, що постійно збільшується породжує проблему автоматичного підведення підсумків. Методологія визначення поглядів та підведення підсумків є важливими прийомами для визначення поглядів. Важливо розуміти, що дані про погляд можуть бути розташовані в тексті, реченні чи області документа, методологія підведення підсумків групує різні погляди в тексті для аналізу сентиментних полярностей, а також степеню появи конкретного погляду в тексті. Для цього можуть використовуватися метод опорних векторів (Support Vector Machine) з лінійним ядром для визначення полярності та ступеня поглядів. Тексти, області тексту, документи оброблюються для пошуку частин з яких можна отримати погляди автора, а згодом для таких частин тексту використовуються методи підведення підсумків (підсумовування) оскільки не всі частини тексту відносяться до даної проблеми, яка розглядається. Дані методи використовуються бізнесу та уряду, так як використання цих

методів допомагає в поліпшенні політики розвику компаній та поліпшенню продуції, послуг і т.д[7].

3. Методологія вилучення поглядів з тексту. Аналіз поглядів використовується для визначення та класифікації особистої інформації в тексті. Ці дані не обов'язково можуть бути визначені як факт, оскільки люди мають різні почуття щодо одного і того самого продукту, послуги, теми або особи. Методологія вилучення поглядів використовується на частинах тексту де наявні дані про погляд користувача, інформація з поглядами може бути проігнорована, якщо вона не націлена на об'єкт, що досліджується.

Реалізація рекомендаційних систем. Соціальні мережі є чинником розвидку напрямку реалізації рекомендаційниї систем. Така система є вбудованою в мережу або є окремим продуктом конкретної компанії. Рекомендаційна система аналізує гетерогенні дані соціальної мережі та надає користувачеві рекомендації щодо, наприклад, друзів чи груп, або товарів, чи навіть показувати рекламу. Уміння робити такі рекомендації є вигідним як користувачу, так і бізнес-компаніям, що продають конкретні товари, а також групам в соціальних мережах, що модуть мати більшу кількість учасників і монетизувати свій контент, продаючи рекламні місця. З моменту введення користувацької інформації в інтерфейс соціальної мережі, користувачеві нажаються різні рекомендації та може бути показано рекламу. Значна частина переходів користувачів до сайтів продажу товарів і т.д. є прямим результатом автоматичних рекомендацій, отриманих з контексту соціальних мереж[7].

Етнографія та онлайнографія. Етнографія – антропологія, що надає науковий опис людських спільнот, в той час як онлайнографія – це ринкові медіа дослідження, що в основному використовують соціальні онлайн дані, побудовані на етнографії. Онлайнографія застосовується для розуміння соціальної взаємодії в контексті цифрових комунікацій. Вона визначається як специфічний набір дослідницькиз практик, пов'язаних зі збором даних, аналізом та етикою

досліджень. Прикладом таких досліджень є, наприклад, дослідження еволюції культури користувачів соціальної мережі YouTube Канзаського Національного університету. Дані таких досліджень можуть бути використані для кращого розуміння інформації, що може бути використана для досліджень у інших сферах, розуміння культури користувачів, а також демографічних показників, наприклад, цікавість певними типами товарів в різних країнах, реакції на події в різних регіонах і т.д[7].

Прогностичний аналіз. Використання методів прогнозтики та штучного інтелекту для прогнозування. В основному використовується для великих компаній для поглиблення взаємодії з клієнтами, оптимізування бізнес процесів та зменшення витрат[7]. Комбінація реальних потоків даних та прогностичної аналітики породжує процес, в якому бізнес може отримувати дані статистики на майбутнє. Прогностична аналітика для великих даних – це нова область, стимульована досягненнями в розвидку потужностей комп'ютерів, технологій баз даних та інструментів та методів для обробки великих даних. Прогностичний аналіз, включаючи цей набір технологій, дозволяє організаціям використовувати збережені дані для переміщення від історичного, описового вигляду до майбутніх перспектив.

В даній роботі буде детально розглянуто основні методи аналізу тексту та вибору основних частин на основі обробки так званої природної мови, а також детально розглянуто сам принцип прогностичного аналізу та його основні методи та використано їх в практичній частині роботи.

1.3 Характеристика даних соціальних мереж та проблема їх обробки

Хоча доступ до даних соціальних мереж доступний з використанням API, через комерційну цінність даних, більшість основних джерел, таких як Facebook і Google не відкривають доступ до своїх "сирих" даних; дуже мало джерел соціальних даних надають такі дані у відкритий доступ. Служби новин, наприклад, Thomson Reuters та Bloomberg стягують плату за доступ до своїх

даних. В той час як Twitter відкриває гранти на дослідження своїх даних для науковців в некомерційних цілях. Науковці мають доступ до даних з 500 мільйонів твітів в день. Дослідники постійно знаходять нові джерела даних для об’єднання та аналізу. Тому, коли використовується текстовий аналіз, потрібно оброблювати різні ресурси, наприклад, RSS-канали, блоги, новини, соціальні мережі, доповнені талакомунікаційними даними, геопросторовими даними, відео даними і т.д. Використання декількох типів даних веде до успішного аналізу. Загалом можна розрізнити такі типи даних – історичні набори даних, а саме попередньо накопичені та збережені новини, фінансові та економічні дані, канали “живих” даних – потокові дані з соціальних мереж типу Twitter, різних ЗМІ. Такі типи даних в основному надходять в необробленому вигляді, в форматі, який надає конкретна соціальна мережа, тому потрібно використовувати засоби для дообробки аби виділити ключову інформацію, необхідну для дослідження. Розглянемо основні проблеми та виклики обробки даних із соціальних мереж.

Різноманітність форматів даних. Дані в соціальних мережах в основному зображаються за допомогою спеціальної мови розмітки HTML (Hypertext Markup Language), адже соціальні мережі розгортають на окремих серверах і більшість користувачів відкривають їх за допомогою веб-браузера. HTML – широко відома мова розмітки веб-сторінок, використовується для перегляду сторінок та розмітки в веб-браузері. HTML складається за HTML елементів, таких як теги, дужки, кутові дужки і т.д., що разом формують вміст веб-сторінки будь-якої соціальної мережі. Деякі сайти новин дозволяють завантажувати свої дані у форматі XML (Extensible Markup Language) – мова розмітки для структурування текстових даних з використанням спеціальних тегів. Більшість відомих соціальних мереж дозволяють отримати дані у форматі JSON використовуючи спеціальний веб інтерфейс, JSON (JavaScript Object Notation) – це відкритий стандарт призначений для серіалізування даних, використовується для обміну даних між серверами, пристроями. Така різноманітність форматів створює необхідність в створенні спеціальних програм-обробників, або так званих парсерів даних, які можуть

обробити отримані дані та видати сирі дані для подальшої обробки текстовими аналізаторами.

Очистка даних від “шуму”. Навіть після попередньої обробки даних парсером, отримані дані ще не можна використовувати для аналізу. Через гетерогенну природу даних соціальних мереж, а також різноманітність цих даних, в основному вони зберігають непотрібну для досліджень інформацію. Видалення ненормованого тексту та непотрібних даних все ще є викликом для аналітиків та науковців. Традиційний підхід до очищення текстових даних - витягування даних в електронну таблицю для подальшого переформатування тексту. Наприклад, Google Refine – це автономний додаток для очищення даних і перетворення в різні формати. Вирази трансформації написані в формі Google Refine Expression Language (GREL) або JYTHON (реалізація мови програмування Python, написана на Java) [6].

Неструктурованість даних. Важливим принципом в обробці та аналізі даних є “якість проти кількості” даних. Насправді, багато деталей про аналітичні моделі визначаються за типом і якістю даних. Природа даних також матиме вплив на базу даних та апаратне забезпечення. Природно, неструктуровані текстові дані можуть бути дуже нецілісніми, брудними. Отже, очищення даних є важливою сферою аналізу соціальних мереж. Процес очищення даних може передбачати видалення типографських помилок або підтвердження та корегування значень знаючи відомий список входжень. Зокрема текст може містити помилкові слова, цитати, програмні коди, зайві пробіли, додаткові рядки, переривання, спеціальні символи, іноземні слова і т. д. Для того, щоб досягти високоякісної обробки тексту, необхідно провести очищення даних. Переглянувши види та джерела сирих даних, ми можемо перейти "очищення" даних для видалення неправильної, непослідовної або відсутньої інформації. Перед обговоренням стратегії очищення даних, важливо визначити можливі проблеми з даними:

1. Відсутні дані – наявна певна частина інформації, яку було включено з певної причини в необроблені дані, які було отримано. Проблеми виникають з числовими даними, коли порожні символи заміняються на 0, які потім вибираються, як наприклад, ціна, а також з текстовими даними - коли відсутнє слово може змінити повне значення речення.
2. Неправильні дані – частина інформації може бути неправильно зазначеною (наприклад, десяткові помилки в числових даних або неправильне слово в текстових даних) або неправильне тлумачення (наприклад, система, яка приймає валютну вартість в $, якщо насправді було передано значення в £, або припускається, що текст використовує діалект англійської США, а не британську англійську).
3. Невідповідні дані – коли є частина інформації, яку було вказано непослідовно. Типовим прикладом є помилки з використанням числових даних, а саме різних форматів дати: 10/04/2014, 14/10/2012 або 14/10/2012. Для текстових даних – використання одного й того ж слова в різних випадках, міксування Української та Польської і т.д.

Оскільки більшість даних про соціальні медіа створюється людьми і тому вона є неструктурованою (тобто вона не має попередньо визначеної структури або моделі даних), потрібно використання алгоритмів, що здатні перетворити такі дані в структуровані, для подальшого аналізу отриманих даних. Тому неструктуровані дані повинні бути попередньо оброблено, теговано, щоб аналізувати дані соціальних мереж. Додавання додаткової інформації до даних (тегування) можна виконувати вручну або за допомогою спеціальних інструментів, які шукають шаблони або інтерпретують дані за допомогою методів такі як аналіз даних та аналіз тексту. Тегування неструктурованих дани зазвичай включає тегування даних за допомогою метаданих. Зрозуміло, що неструктурований характер даних соціальних мереж призводить до неоднозначності і нерівномірності, після обробки комп’ютером. Використання єдиного набору даних може дати цікаві результати.

Недоступність “сирих” даних. Дані соціальних мереж є захищеними і в основному доступ до відкритих даних надається через API, тобто дослідники не мають доступу до відкритих баз даних для виконання аналізу. Тому потрібно використовувати спеціальні бібліотеки для обробки HTTP запитів на сервери соціальних мереж для отримання порції даних, при чому, через гетерогенність та постійну оновлюванність даних, цей процес стає набагато складнішим. Деякі соціальні мережі навіть не надають доступу до своїх даних через API інтерфейси безкоштовно, або вимагають використання спеціалізованих інструментів - джерела, які забезпечують контрольований доступ до своїх даних про соціальні медіа через спеціальні інструменти, задля полегшення отримання таких даних, так і для зупинення можливості викрадення даних[6]. Їх можна розділити на:

1. Безкоштовні джерела – репозитарії, які є вільно доступними, але захищені, або можуть обмежувати доступ до так званих “сирих” даних.
2. Комерційні джерела – реселлери даних, які стягують плату за доступ до своїх даних з соціальних мереж.

Існує все більша кількість комерційних сервісів, що оброблюють дані соціальних мереж, задля забезпечення платного доступу за допомогою простих інструментів аналітики. Крім того, компанії, такі як Twitter, обмежують вільний доступ до своїх даних та ліцензіюють свої дані реселлерам, таким як Gnip та DataSift. Gnip є найбільшим постачальником таких даних у світі.

1.4 Висновки за розділом

У даному розділі було розглянуто та поняття гетерогенних та гомогенних джерел даних та підкреслено їх основні відиінності. Описано поняття соціальних мереж, що включає не лише відомі веб-сайти такі як Twitter чи Facebook, а й RSS, блоги, стрічки новин. Оглянуто основні типи соціальних мереж – публічні та закриті, а також їх види – інформаційні, аматорські, пошуку роботи та канддатів, дослідницькі та освітні. Було продемонстровано основні області досліджень соціальних мереж та проблеми в обробці даних таких джерел як соціальні мережі.

Як видно, соціальні мережі представляють собою цікавий предмет досліджень, де інформація оновлюється з надзвичайною швидкістю, саме соціальні мережі є основними джерелами даних для так званих рекомендаційних систем, а дослідження в області соціальних медіа даних дозволяють використовувати отриману інформацію у різних сферах життя – дані, доступні в соціальних мережах, можуть дати уявлення про людину, громади та суспільство в цілому, що раніше було неможливим в таких глобальних масштабах. Такі цифрові медіа дані виходять за межі фізичного світу для вивчення людських відносин і допомагають визначати популярні соціальні та політичні настрої для регіональних груп без використання опитувань. Соціальні мережі регулярно фіксують основні маркетингові тренди, настрої та напрями та є ідеальними джерелами для вивчення та обробки.

В наступному розділі будуть зазначені основні алгоритми та методи прогнозування, що будуть використані в практичній роботі, а також інструменти, що буде використано для обробки даних із гетерогенних джерел.

2 ОГЛЯД МЕТОДОЛОГІЇ ОБРОБКИ ДАНИХ З ГЕТЕРОГЕННИХ ДЖЕРЕЛ

2.1 Алгоритми та методи прогнозування

Для того, аби виконати прогнозування популярності та використовуваності фреймворків та мов програмування будемо використовувати засоби та алгоритми машинного навчання на отриманих даних з соціальних мереж пошуку роботи. Отримані дані будуть подаватися у вигляді пари ключ-значення “назва фреймворка/мови програмування” – “кількість запитів для кандидатів”. За такими даними можна побудувати аналітичну модель, а з її використанням отримати прогнозовані результати щодо популярності та використовуваності даного фреймворка. Розглянемо основні алгоритми, що будуть протестовані в практичній частині роботи.

Найпростішим алгоритмом що походить із області статистики є метод лінійної регресії. З використанням лінійної регресії ми прогнозуємо оцінку однієї змінної що залежить від значень іншої. Значення, що дослідник хоче отримати називається змінною критерію і зачасту позначається символом “Y”. Змінна, на основі якої виконується аналіз позначається символом “X”. При виконанні лінійної регресії зв’язок між даними моделюється з використанням так званих лінійних функцій, в той же час, невідомі параметри моделі оцінюються за вхідними даними моделі. Подібно до інших методів регресійного аналізу, що будуть розглянуті далі, лінійна регресія повертає розподіл ймовірності змінної Y в залежності від значення змінної X, а не розподіл їх спільних ймовірностей. При прорахунку таких моделей як правило використовується так званий метод найменших квадратів, завдяки якому корегується помилка, проте можуть використовуватися й інші методи[11].

Загалом лінійну регресію можна позначити у такому вигляді:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \varepsilon \quad (2.1)$$

Де y – залежна змінна, а набір x – відповідно незалежні пояснювальні змінні, ε – похибка моделі. В той же час математичне сподівання для даної змінної є також лінійною функцією і розраховується за формулою:

$$E(y) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \varepsilon \quad (2.2)$$

Вектор параметрів β не є відомим і задачею лінійної регресії є оцінка даних параметрів на основі вже існуючих наборів даних для y та x. Тобто на основі n-ного числа експериментів є відомими значення $\{y_i, x_1 \dots x_n\}_{i=1}^{n}$ незалежних змінних та відповідне значення залежної змінної.

В результаті отримаємо матричне рівняння для змінної y, враховуючи кількість експериментів:

$$y = X\beta + \varepsilon \quad (2.3)$$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix},\ X = \begin{pmatrix} 1\ x_{11} \dots\ x_{1k} \\ 1\ x_{21} \dots\ x_{2k} \\ \dots \\ 1\ x_{n1} \dots x_{nk} \end{pmatrix},\ \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_n \end{pmatrix} \quad (2.4)$$

На основі цих даних відповідно необхідно оцінити значення параметрів β, а такод розподіл похибки ε. Для того аби мінімізувати похибку зачасту використовується метод найменших квадратів. Цей метод приймає за оцінку параметра значення, що мінімізують суму квадратів залишків по всім спостереженням:

$$\hat{\beta} = argmin \sum_{i=1}^{n} |y_i - \beta_0 - \sum_{j=1}^{k} x_{ij}\beta_j|^2 = argmin||y - X\beta||^2 \quad (2.5)$$

В результаті маючи отримані дані можна виконати прогнозування для таких даних в майбутньому. Розглянемо приклад отриманої статистики деякого фреймворка за певну кількість днів (хоча це і малий проміжок для аналізу) в вакансіях пошуку кандидатів де по осі X – гіпотетичний день і по осі Y – кількість вакансій по заданому фреймворку в цей день як на рис. 2.1:

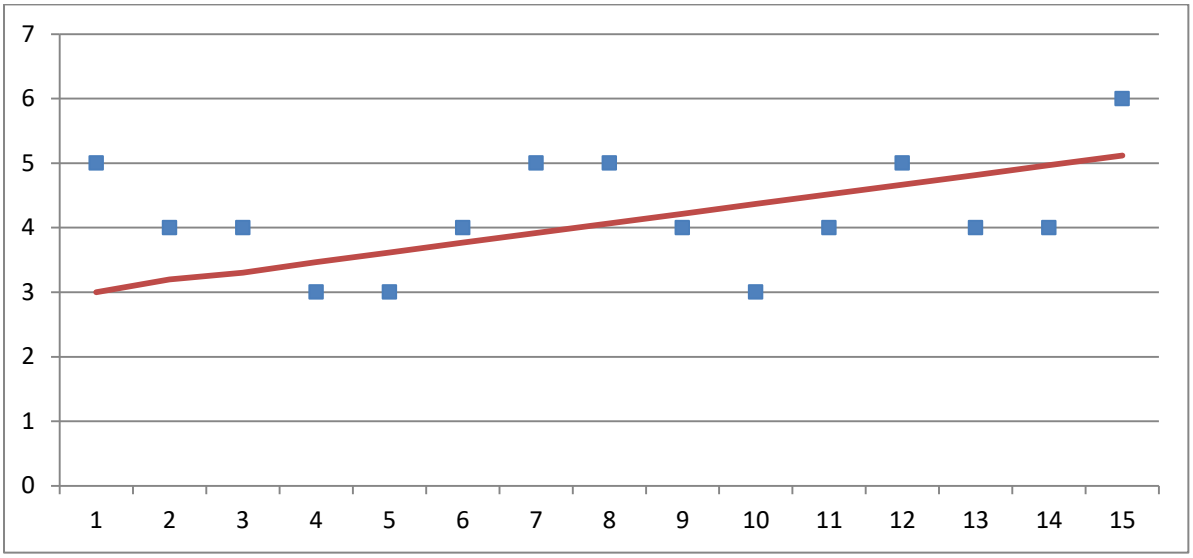

Рисунок 2.1 – Приклад моделі лінійної регресії

В результаті отриманих даних можна побудувати пряму і мінізувати її похибку на основі методу найменших квадратів. На графіку її позначено червоним кольором, якщо від кожної точки з даними провести перпендикуляр на таку пряму, то сума таких результатів повинна бути найменшою, тоді наша модель може в принципі правильно прогнозувати наступні показання. З малюнка видно, що така пряма ще не є правильною, проте похибку можна мінімізувати і в принципі отримати необхідні результати, на графіку пряма “росте” з часом, що не є правильним в даній ситуації, і кут повинен бути меншим. Недоліком такого методу є його лінійність, хоча дані не завжди піддаються такій природі.

Ще одним регресійним методом, що можна використати для прогнозування є так звана квантильна регресія, яка має певні переваги над звичайною лінійною регресією проте це не значить, що звичайну лінійну регресію не потрібно використовувати, в деяких випадках простота є більгшим пріорітетом і явна лінійна характеристика даних може дати гарні результати при використанні лінійної регресії. Загалом, лінійна регресія підсумовує зв’язки між набором регресорів а значення вихідної змінної Y базується на значенні функції $E(y|x)$ (формула 2.2). Однак такий підхід дозволяє нам отримати лише часткове

представлення залежності, а нам би хотілося отримати залежності на різних точках Y. Саме це дозволяє нам зробити застосування квантильної регресії.

Аналогічно до використанної функції в лінійній регресії ми хочемо отримати значення використовуючи, як приклад, медіанну функцію $Q_\tau(y|x)$ де медіана це 50% або квантиль τ всього набору. Припустимо, нам би хотілося отримати значення не лише даних використання фреймворку в залежності від року, але й в залежності від компанії, при чому нам би хотілося знайти зв'язок від кількості людей у компанії (розмір компанії – більше 3000 працівників, більше 1000, до 100 і т.д.) і потреби у працівниках, що знають конкретний фреймворк. Квантильна регресія – ось, що може допомогти в такій ситуації, розбивши дані на квантилі 0.1 – до 100, 0.5 – 1000 працівників і т.д., звичайно розподіл може бути і не таким. Для цієї задачі нам потрібно мінімізувати:

$$min \sum_{i=1}^{n} \rho_\tau(y_i - \varepsilon) \text{ (2.6)}$$

Використовуючи формулу для лінійної регресії 2.5 та формулу 2.6, приймаючи квантильну функцію:

$$Q_Y(\tau|X = x) = x_i'\beta(\tau) \text{ (2.7)}$$

Отримаємо:

$$\hat{\beta}(\tau) = \ argmin \ \sum_{i=1}^{n} |\rho_\tau(y_i - x'\beta)| \text{ (2.8)}$$

Принцип даного методу є тим же, що і в звичайній лінійній регресії, проте даний алгоритм дозволяє розбивати дані на квантилі різних порядків і знаходити залежності в різних квантилях. Медіанна регресія є підтипом квантильної регресії, де медіана є квантилем $\tau = 0.5$. Проте, хоча дані методи дають непогані результати і в рамках даної роботи вони є ідеальними кандидатами, оскільки розподіл використання фреймворків, мов програмування, інструментів розробки з часом, проте потрібно зазначити і інші методи прогнозування, що можна використати в рамках даної роботи.

Ще одним способом для прогнозування є використання нейронних мереж. Дослідження в області нейронних мереж є результатом досліджень спеціалізованих клітин названих нейронами. Нейрон – клітина, що має декілька входів що можуть бути в стані активації за рахунок виникнення деякого зовнішнього процесу. В залежності від кількості активацій а також ваги активацій, нейрон виробляє свою власну вагу активації та відправляє її на свої виходи. Крім того, конкретні входи чи виходи можуть бути зміцнені, тобто вони мають більшу вагу ніж інші входи чи виходи з даного нейрону. Гіпотеза того, що людський мозок це не що інше, як мережа нейронів дозволяє нам емулювати мозок змоделювавши нейрон і включити таки нейрони в мережу через граф з різними вагами на ребрах[12].

Штучний еквівалент нейрона – це так званий вузол (його також називають нейроном, проте в даній роботі буде використовуватися термін вузол, щоб не викликати плутанини), який отримує набір входів, що мають певну вагу, даний вузол розраховує їх суму за допомогою функції активації φ і передає результати даної функції до наступних вузлів у графі. В загальному вигляді можна представити це за допомогою формули:

$$\varphi(\sum_i \omega_i a_i) = \varphi(\omega^T a) \quad (2.7)$$

Візуально даний вираз можна представити так як на рис 2.2:

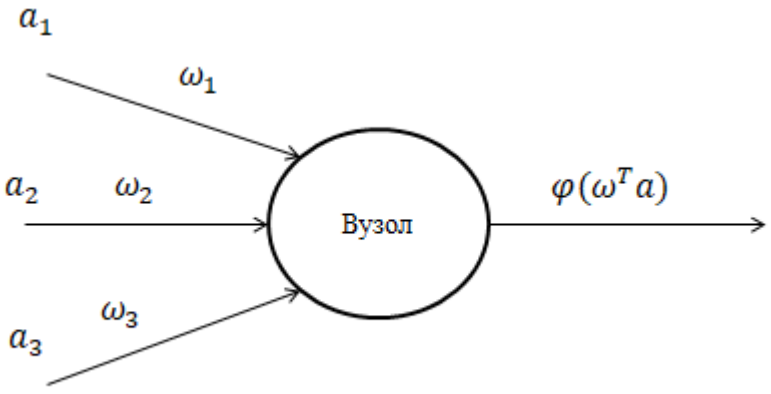

Рисунок 2.2 – Вузол нейронної мережі

При чому основною функцією активації є лінійна:

$$\varphi(\omega^T a) = \omega^T a \quad (2.8)$$

Ще однією популярною функцією є сигмоїда:

$$\varphi(\omega^T a) = \frac{1}{1+e^{-\omega^T a}} \quad (2.9)$$

Широко використовуваною функцією є також гіперболічний тангенс:

$$\varphi(\omega^T a) = \tanh(\omega^T a) \quad (2.10)$$

В результаті можна сформувати мережу, об’єднавши ці вузли разом. Зазначимо, що при використанні мереж для виконання операцій прогнозування простого створення такої мережі може бути недостатньо, така мережа може бути використана для виконання операції регресії, однак для виконання операції прогнозування необхідно використовувати так звані рекурсивні мережі, де попередній вихід може бути з‘єднано з наступним входом як позначено на 2.3:

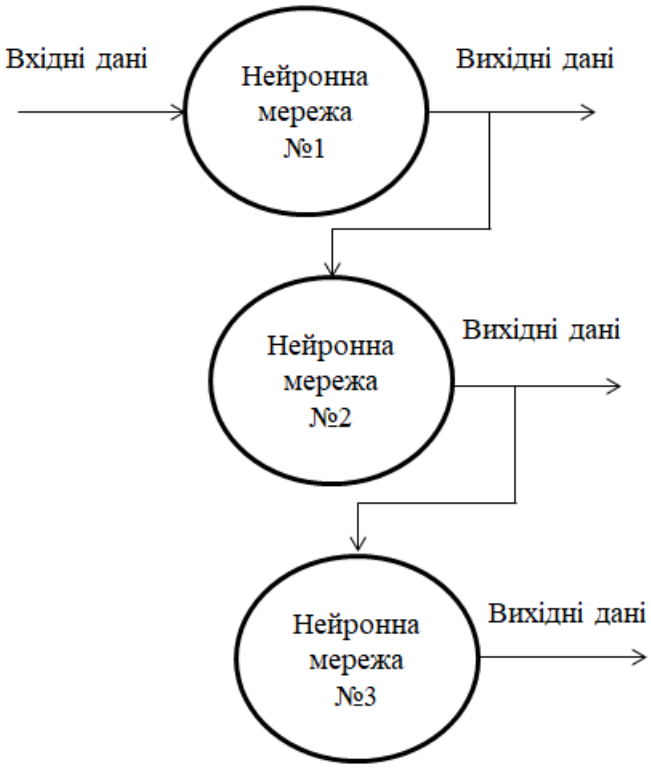

Рисунок 2.3 – Проста рекурсивна нейронна мережа

Хоча на рисунку 2.3 нарисовано лише декілька нейронних мереж, об’єднаних в одну, а саме три, потрібно зазначити, що при виконанні реальної задачі таких мереж може бути більше, також на рисунку 2.3 не зазначена точна кількість необхідних внутрішніх шарів такої мережі. В результаті можна уявити, як повинна працювати така мережа, при подачі на неї значень попередніх уже відомих інтервалів, перепишемо формулу 2.10 і отримаємо [20]:

$$\varphi_t(\omega^T a) = \tanh(\varphi_{t-1}(\omega^T a) + u^T x_t) \quad (2.11)$$

Де змінна u позначає значення ваги при вході на мережу в зазначеній рекурсивній, а коефіцієнт x – відповідне задане значення при вході. Зазначимо, що використанням такої мережі можна виконувати прогнозування на задані короткі проміжки часу [20].

Для того, щоб виконувати прогнозування на більші терміни, можна використовувати LSTM (Long Short-Term) нейронну мережу, що є покращеною рекурсивною мережею, яка може бути складеною з декількох нейронних мереж, кожна з яких використовує внутрішні змінні станів, що передаються між вузлами і відповідно модифікуються [20].

Затвор забуття. Його основна задача – змінювати внутрішній стан мережі. Такий затвор представляється у вигляді функції, він може приймати поставлені значення входів мережі та інші необхідні значення, з яким потім використовується операція множення значення внутрішнього стану, при виході 0 із затвору, внутрішній стан стає приймає значення 0, саме тому його називають затвором забуття. Формула такого перетворення [20]:

$$f_t = \sigma\left(\omega_f[h_{t-1}, x_t] + b_f\right) \quad (2.12)$$

Вхідний затвор повинен використовувати такі значення як вихідне з мережі, що використовується перед заданою, вхідні значення, видаючи на виході число з діапазону до 1, який використовується для перерахунку стану [20]:

$$i_t = \sigma(\omega_i[h_{t-1}, x_t] + b_i) \quad (2.13)$$

Формула внутрішнього стану:

$$C_t = f_t C_{t-1} + i_t C_t \quad (2.14)$$

Для контролю відповідних значень на виході та коефіцієнта внутрішнього стану використовується вихідний затвор, в результаті така частина внутрішнього стану передається на вихід [20]:

$$O_t = \sigma(\omega_o[h_{t-1}, x_t] + b_o) \quad (2.15)$$

$$h_t = O_t \tanh(C_t) \quad (2.16)$$

Таку модель нейронної мережі можна використовувати для прогнозування на великі проміжки даних на відміну від раніше описаної моделі з декількома мередами об’єднаними в одну рекурсивно [20].

Найпростішим методом прогнозування є метод, що має назву Exponential Smoothing (експоненційного згладжування). Даний метод дозволяє виконувати прогнозування короткі проміжки, як, наприклад, на наступний тиждень чи наступний день чи рік, проте його можна використовувати і на більші проміжки часу, однак результат матиме ще меншу точність [8, 20].

Приймається, що дані за певний період часу t позначаються x_t, прогноз для певного часу буде позначатися як s_t. Основну формулу даного методу можна записати так [20]:

$$s_t = \alpha x_{t-1} + (1 - \alpha) s_{t-1} \quad (2.17)$$

Можна зробити висновок, що основні значення прогнозів на наступних кроках мають залежність від попереднього значення а також попереднього прогнозу. Також прогнози для уже відомих значень приймаються як рівними таким, а при неоюхідності прогнозування на більші інтервали, то приймається, що отримане прогнозоване значення є реальним і є очікуваним, стандартним значенням α зачасту обирається значення 0,5 [20].

Даний алгоритм можна назвати експоненційним оскільки він має експоненційний характер, при розписі формули в повному вигляді [20]:

$$s_t = \alpha[x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + \cdots + (1-\alpha)^{t-1}x_1] + (1-\alpha)^t x_0 \quad (2.18)$$

Незважаючи на те, що при використанні даного алгоритму можна отримати непогані результати, всеж потрібно зазначити, що в основному використовуються покращені версії алгоритму, зазначені далі.

Введемо наступний параметр b_t – найкраща оцінка для даного інтервалу в часі, саме так і працює алгоритм другого порядку, а в результаті буде отримано формулу 2.17:

$$s_t = \alpha x_{t-1} + (1-\alpha)(s_{t-1} + b_{t-1}) \quad (2.19)$$

$$b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1} \quad (2.20)$$

З використанням попередньої формули дані будуть мати більш точний характер прогнозу при порівнянні з попередньою формулою, описаної для менших порядків[9, 20].

Останнім способом прознозування є використання ARIMA моделей. Такі моделі, теоретично, є загальними що використовуються у прогнозуванні, у поєднанні з так званими перетвореннями нелінійних величин. Випадкова величина, що є часовим рядом є стаціонарною, якщо відповідні її властивості змінюються, такі серії не мають спеціальних тенденцій, а вони варіюються навколо свого середнього значення. ARIMA (Auto-Regressive Integrated Moving Average) використовує основні три компоненти описані далі [13, 20].

Авторегресивний компонент використовує попередні значення в регресійному рівнянні серій Y. У визначенні $ARIMA\ (p, d, q)$ параметр p зазначає кількість попередніх результатів, що будуть використані у рівнянні, а сама модель авторегресії позначається як $AR(p)$. Для прикладу, модель $AR(2)$ або $ARIMA(2, 0, 0)$ буде виглядяти так [20]:

$$Y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + e_t \quad (2.21)$$

Ще одним компонентом такої моделі є так званий компонент інтегрування або інтегрований $I(d)$ компонент, параметр d даного компоненту відповідає за ступінь різності в даній моделі. Тут різниця між серіями передбачає в собі виконання простої операції віднімання значень теперішнього і попередніх значень d разів [20].

Останнім компонентом ARIMA моделі є так зване переміщення середнього значення або середнє переміщення, що позначається як $MA(q)$ та відповідає за помилку моделі з використанням комбінації попереднії помилок, позначених як e_t. Параметр q відповідає за прийняте значення помилок у необхідній моделі. В результаті, можна отримати таку формулу [20]:

$$Y_t = c + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} + e_t \quad (2.22)$$

При поєднанні усіх трьох моделей на вихід отримаємо ARIMA модель для розрахунку, така модель має вигляд лінійного рівняння, що має наступний вигляд як у формулі 2.23 [20]:

$$Y_t = c + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t \quad (2.23)$$

ARIMA має свої переваги та недоліки, так наприклад, ARIMA є лінійною проте для її використання необхідно мати набір із попередніх значень, і хоча, на відміну від лінійної регресії, значення, що будуть прогнозуватися не обов'язково повинні мати лінійний характер, однак, хоча основною ціллю є прогнозування значень у моделі, використання ARIMA не дасть причини того, чому буде таке значення і залежність параметрів моделі.

Отже, основними алгоритмами та методами для прогнозування, що будуть використані в практичній частині було обрано такі:

1. Лінійна регресія та вдосконалена ординальна лінійна регресія;
2. Рекурсивна нейронна мережа;

3. ARIMA модель;
4. ES алгоритм.

2.2 Вибір моделі обробки великих масивів даних

Оскільки дані з соціальних мереж мають гетерогенний характер, а також оновлюються з надзвичайною швидкістю, то для обробки таких великих масивів даних потрібно використовувати спеціальні інструментарії перевірені часом, що зарекомендували себе як швидкі та надійні, одною з таких технологій є використання підходу MapReduce. Потрібно зазначити, що в даному контексті MapReduce розглядається не як конкретна реалізація, як наприклад, Hadoop MapReduce, а підхід для обробки великих масивів даних. Розглянемо основні принципи такого підходу.

На вхід функції Map необхідно передавати різного типу даних, а саме масиви даних, різного типу документи, де кожна така частина є сукупністю елементарних часточок даних. З'єднані виходи із операції Map та входи операції Reduce в результаті обробки будуть представлятися у вигляді пар ключ-значення, така подача входів та виходів є спеціальною для обробки з використанням MapReduce. Для прикладу, описана раніше функція Map може приймати на вхід деяке значення, для прикладу частину раніше оброблених якимось сервісом даних, що була задано, видає на вихід, в залежності від ситуації, список пар ключів та значень, або у випадку пустної множини – пусту пару ключів та значень. Важливим аспектом MapReduce є те, ключі повинні бути однаковими, це розумне зауваження, оскільки не маючи різних ключів, операція не зможе бути виконаною, оскільки групування буде не точним і не відповідатиме реальним даним, самі ж дані в групі можуть мати повторюючі значення. Зазначимл, що кожен з вузлів операцій Map і Reduce може бути використано для потокової обробки даних на вхід, тобто, після обробки одного пакету даних та генерації потоквого виходу, такий вузол може прийняти та обробити необхідні дані [20].

Після такої обробки основний контроллер системи запускає необхідні задачі на виконання в результаті чого, будуть отримані такі пари ключів та значень як на 2.24 [20]:

$$\ll K_1, V_1 >, < K_2, V_2 >, < K_1, V_3 >, \dots, < K_N, V_N \gg \quad (2.24)$$

Перетворюються в груповані пари виду:

$$< K_1, < V_1, V_3, \dots \gg, \dots, < K_N, < V_N, \dots \gg \quad (2.25)$$

Після даного перетворення використані дані необхідно направляти на обробку хендлерів операції Reduce. На виході операції Reduce, як і в використанні Map, отримаємо пари ключ-значення, при чому ключі та значення можуть мати інший тип, в залежності від поставленої задачі, проте в більшості випадків вони матимуть однаковий тип. Усі такі пари з усіх вузлів будуть згруповані в один файл, що і є результатом цієї операції. Зазначимо, що для виконання завдання може бути недостатньо виконання лише однієї команди MapReduce, тому вихідні значення однієї з операцій MapReduce можуть бути вхідними параметрами для деяких інших операцій MapReduce, поставлених в чергу. На рис. 2.1 зображено описаний принцип роботи MapReduce [20].

Хоча вибір основного підходу для обробки великих масивів даних було обрано, потрібно обгрунтувати та пояснити основні недоліки в порівняні з іншими підходами та можливості, які вони надають.

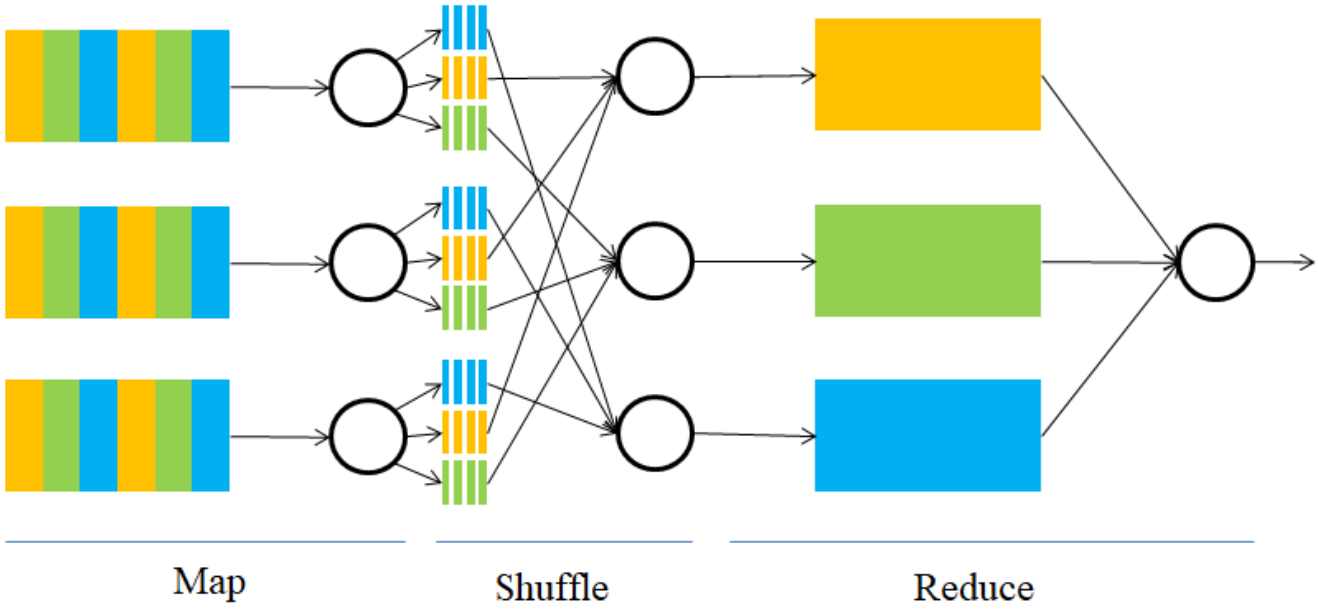

Рисунок 2.4 – Принцип роботи MapReduce

Ще одним способом є вискористання реляційної бази даних для збереження даних на обробку та серверів для виконання обчислень з реляційними даними. В більшсті випадків можна розглядати принципи MapReduce як доповнення до СУБД. Перевагу СУБД слід надати при роботі з одиночними запитами, де дані доставляються з невисокою частотою, проте зчитування відбувається часто. За таким принципом працює більшість веб-додатків, в яких операції додавання даних є значно рідшими ніж їх зчитування, наприклад, інтернет магазин, в якому більшість клієнтів спочатку проходять по каталогу товарів, і обираючи необхідний, починають відправляти необхідні дані про бажану покупку на сервер інтернет магазину. Проте, для обчислень з великими масивами даних, такий підхід не є оптимальним, оскільки відбувається велика кількість запитів на запис таких даних, а саме зчитування відбувається лише для обробки. З іншого боку, для оновлення даних в реляційних базах використовуються бінарні дерева пошуку, проте показано, що принципи MapReduce з використанням Sort-Merge є більш ефективними для обробки наборів даних з різних гетерогенних джерел, що постійно оновлюються. Ще однією проблемою використання реляційних баз є їх транзакційна природа, що може значно зповільнити обробку великих масивів даних при частому записі таких даних у базу. Однак, деякі бази даних, такі як Greenplum почали використовувати ідеї MapReduce для своїх імплементацій. На

основі даних недоліків реляційних баз даних їх використання в практичній частині даної роботи не є доцільним.

Ще одним підходом для обробки таких даних є використання технології Message Passing Interface (MPI) на основі Grid обчислень. Такий підхід чудово зарекомендував себе для виконання паралельних обчислень, проте існують недоліки з доступом до великих обсягів даних кожного вузла, де пропускна здатність мережі є низькою і обчислювальний вузол переходить у стан простою. В основних реалізаціях підходу MapReduce дані знаходяться чи на жорсткому диску вузла чи в його оперативній пам’яті, тобто всі MapReduce реалізації дотримуються того, що пропускна здатність мережі є цінним ресурсом при виконанні обчислень. Ще одним недоліком MPI є його складність, при використанні реалізацій MapReduce все, що потрібно зробити програмісту – реалізація відповідних методів Map та Reduce, а за все інше опікується фреймворк, при використанні MPI програміст повинен контролювати процеси обробки потоку даних, доступність вузлів, розподілення даних між вузлами, звісно, такий підхід надає більшу свободу для конкретної задачі, проте відволікає від фокусування для написання логіки обробки даних для задачі. Хоча MPI і виграє в швидкості при обробці даних на конкретному вузлі, проте швидкість зчитування, важкість в написанні та необхідність розгортання таких вузлів на власному кластері, все ж заставляє використовувати MapReduce підхід.

Основною реалізацією підходу MapReduce є фреймворк Apache Hadoop. Hadoop – це програмна інфраструктура з відкритим вихідним кодом, яку можна встановити на кластері машин для спільного зберігання даних великиго об’єму та їх подальшої обробки на кластері. До складу Hadoop входить ціла інфраструктура програмного забезпечення для виконання таких обчислень з використання серій виконання Map та Reduce, як вже було зазначено, Hadoop бере на себе всі задачі з розподілення, збереження даних при виконанні обчислень, а все, що потрібно від розробника – реалізація Map та Reduce.

Головними перевагами Hadoop MapReduce над попередніми підходами є широкий спектр підтримки мов програмування, а саме Java, C++ чи Python, на відміну від того ж MPI, для Hadoop реалізовано спеціальні трекери задач та балансувальники задач, що дозволяють рестартувати задачу при виникненні помилок без впливу на інші процесу у системі чи інші задачі, також Hadoop дозволяє приорітизувати задачі та виконувати спочатку лише задачі, що оброблюють основний потік даних. Завдяки своїй гнучкості, Hadoop дозволяє оброблювати дані різних типів та структур, а також створює копії даних для обробки в разі виникнення помилок чи втрати даних.

Основними компонентами Hadoop є HDFS та YARN. HDFS – Hadoop Distributed File System була розроблена з використанням архітектури розподіленої файлової системи. Вона працює на апаратному забезпеченні. На відміну від інших розподілених систем, HDFS є надзвичайно нестабільнозахисною та працює з використанням недорогих апаратних засобів, HDFS може зберігати дуже великий обсяг даних і забезпечує простий доступ до них. Щоб зберігати такий величезний обсяг даних, файли зберігаються на декількох машинах. Ці файли зберігаються з резервними копіями, щоб запобігти від можливих втрат даних у випадку помилки. Завдяки цьому HDFS дозволяє оброблювати таку кількість даних паралельно. HDFS також включає в себе систему знаходження помилок та їх виправлення, дозволяє розміщати тисячі вузлів у кластері та розподіляти дані між ними, а завдяки такому розподіленню, виконання обробки таких даних виконується дуже швидко, оскільки дані знаходяться на обчислювальних вузлах. В свою чергу, YARN є архітектурним центром Hadoop, що дозволяє використання декількох движків обробки даних, підтримки стрімінгу в реальному часі[15].

Хоча Hadoop має так багато переваг та архітектуру, що чудово підтримує обробку великих масивів даних, все ж даний підхід має декілька серйозних недоліків. Hadoop чудово підходить для файлів з великими розмірами, проте, якщо в наявності є лише файли малого розміру проте у великий кількості, то Hadoop втрачає швидкість обробки через те, що обробка таких файлів потребує

частого зчитування даних з жорсткого диску, на якому вони зберізаються. Для обробки даних в декілька етапів необхідно запускати декілька MapReduce процесів, що додає складності обробки. Всі ці проблеми покликано вирішити з використанням більш сучасного фреймворка Apache Spark, а сам Hadoop для задач обробки великих масивів даних вважають застарілим та нажають перевагу Spark.

Spark – фреймворк, що призначений для розподіленої обробки даних з використанням спеціальних функцій та примітивів, для обробки в оперативній пам’яті, що дозволяє збільшити швидкість пакетної обробки в 100 разів, порівнюючи з Hadoop. Це потужний комплекс засобів з відкритим кодом, легкий в користуванні та інтерфейсами API для Java, Scala, Python, R та навіть SQL. Його можна використовувати для побудови додатків, що оброблюють дані у якості бібліотеки, чи просто у якості засобу для проведення аналізу даних. Spark включає в себе стек бібліотек включаючи бібліотеки для роботи з SQL, DataFrames, MLlib, що використовується для машинного навчання, а також Spark Streaming для проведення аналізу даних на льоту. Spark може працювати на робочій машині, Hadoop, на окремому сервері чи в хмарному середовищі. Spark має спеціальні модулі для доступу до таких джерел даних як HDFS, Apache Cassandra, Apache Hbase та S3. Spark широко використовують такі відомі бренди як Netflix, Yahoo та Tencent, що разом оброблюють декілька петабайтів даних на кластерах, що складаються з 8000 вузлів.

В основі Apache Spark розташована концепція абстракції даних, що є розподіленою колекцією об’єктів і називається Resilient Distributed Dataset (RDD), що дозволяє писати програми оперуючи даними розподіленими наборами даних. Колекції RDD є незмінними, тобто програміст не зможе змінити елементи колекції, розширити колекцію чи видалити елемент з такої колекції, при чому вони репрезентують дані, що можуть бути збережені в пам’яті чи на диску поміж кластеру машин. Дані є розподіленими між машинами на кластері, що дозволяє програмісту оброблювати їх паралельно, використовуючи низькорівневий API

який надає можливості використовувати трансформації та виконувати різні дії з такими даними, також колекції RDD є стійкими до помилок і, якщо з якоїсь причини виник збій чи дані було втрачено, Spark автоматично зробить відкат до копії втрачених даних.

Разом з пакетом Spark розробнику надається спеціальний набір інструментів та алгоритмів машинного навчання в бібліотеці MLlib. MLlib було спроектовано для підтримки простоти, розширюванності і легкої інтеграції з іншими інструментами, що використовує розробник. З такою розширюванністю, підтримкою різних мов, а також швидкістю Spark, розробники можуть вирішувати свої задачі пов’язані з великими масивами даних набагато швидше ніж використовуючи перераховані аналоги. Основною перевагою MLlib є те, що розробникам та вченим не потрібно фокусуватися на проблемах, пов’язаних із важкістю розподілення ресурсів на вузлах, пріорітизацією задач, вирішенню питань із помилками при роботі, а також розподіленню даних та пристосуванню алгоритмів машинного навчання до роботи в розподілених середовищах, все це реалізовано командою Spark. Проте варто зазначити, що в Mllib реалізовані лише основні алгоритми, що підходять для широкого класу задач, проте розробнику надається можливість розширення реалізації цих алгоритмів для своєї конкретної задачі[14].

В пакет Spark входять ще багато цікавих можливостей та інструментів, як Catalyst Optimizer, Spark Streaming, Dataset, GraphX, ML Pipeline і інше, проте в рамках даної роботи вони розглядатися не будуть. Проте навіть маючи так багато переваг, пакет Spark має й свої недоліки, як вже зазначалося, всі операції в Spark використовують оперативну пам’ять, що дозволяє обробляти дані в 100 разів швидше ніж з використанням Hadoop, проте при операціях, що використовують багато пам’яті це може стати проблемою, ще одним недоліком є те, що Spark не надає способу збереження даних та їх обробки на жорсткому диску як Hadoop, що іноді заставляє викоритовувати Hadoop замість Spark[14].

З наведених вище способів обробки великих масивів даних немає однозначного переможця, кожна з технологій має свої переваги та недоліки, пов’язані з часом обробки, затратами пам’яті, використанням ресурсів та коштом апаратного забезпечення, що потрібно використовувати, наприклад, використання SQL бази є найдешевшою альтернативою, проте в той же час є найповільнішим способом. В рамках цієї роботи MPI не буде розглядатись, як одна із альтернатив, оскільки даний спосіб є досить складним у настройці та обробці даних, та й підтримка його хмарними технологіями є нульовою. Двома найкращими варіантами є використання Hadoop та Spark, більше того Google Cloud Engine та Azure надають підписку на використання як Hadoop так і Spark в своїх серверах, що облегшує процес розробки. Однак, оскільки Spark має в своєму пакеті бібліотеку машинного навчання, що може допомогти при виконанні операцій прогнозування та виділення тексту, то вибір стає очевидним, а саме використання Spark як засобу обробки великих масивів даних.

2.3 Вибір архітектури додатку для підтримки усіх процесів

Вибір архітектури є важливим етапом при розробці будь-якого додатку. При розробці даного типу додатку очевидним вибором є клієнт-серверна архітектура, що зарекомендувала себе в продовж останніх десятків років. Практична частина даної роботи не буде виключенням і буде реалізацією саме такого підходу при написанні додатків. Розглянемо основні принципи клієнт-серверної архітектури.

Клієнт-серверна архітектура набула своєї популярності завдяки швидкому розвидку мережі Інтернет а також значному впливу та використанню баз даних для збереження даних. Таку архітектуру можна позначити як концепцію мережі, і якій усі ресурси розгорнуто на серверах, які використовуються для задоволення потреб своїх клієнтів, а основними компонентами такої системи можуть бути компоненти зі списку:

1. сервери, що надають інформацію та інші типи даних додаткам, які звертаються до них;
2. клієнти, які звертаються до відповідниї сервісів;
3. мережа, що забезпечує взаємодію між серверами та клієнтами.

Правила взаємодії між клієнтом і сервером називають протоколи взаємодії між користувачами мережі. Кожен із таких клієнтів може бути як сервером так і клієнтом такого сервера і вои мають свої обов'язки. Можна видокремити основні такі рівні - рівень представлення даних, так званий користувацький інтерфейс з різними компонентами для взаємодії користувача із системою. Прикладний рівень, який реалізує основну логіку додатку Рівень управління даними, що дозволяє використовувати дані та мати до них доступ.

У дволанковій клієнт-серверній архітектурі виконується взаємодія двох основних компонентів клієнта та сервера. Зазначимо, що від використання та присвоювання конкретних функцій для таких клієнтів та серверів можуть розрізняти моделі товстих та тонких клієнтів, де товстими клієнтами можуть бути такі пристої як кишенькові комп'ютери, мобільні телефони та ін[16].

Трьохланкова клієнт-серверна архітектура базується на відділенні рівнів прикладного та управління даними. Створюється спеціальний програмний рівень, в якому зосереджені логіка додатку, програми проміжного рівня функціонують під управлінням спеціальних додатків, проте запуск таких додатків повинен виконуватися з використанням спеціальних веб-серверів. Управління даними здійснюється спеціальним сервером даниї. [16].

Дволанкова архітектура простіша, усі запити оброблюються єдиним сервером, проте через це вона є менш надійною і потребує підвищених вимог до продуктивності сервера. Трьохланкова архітектура складніша, але завдяки тому, що функції розподілені між серверами другого і третього рівня, ця архітектура проявляє[16]:

1. гнучкість, а також масштабованість;

2. високий рівень безпеки;
3. Високу продуктивність.

Хоча трьохланкова архітектура і дозволяє створити досить гнучкі додатки, проте для даного типу додатку вона не підходить з ряду причин, по-перше обробка тексту повинна відбуватися у два етапи, а саме, виділення необхідної інформації з отриманих даних соціальної мережі, а згодом їх обробка для виконання прогнозування та виводу результатів користувачу, більше того така операції повинні бути виконані окремо, при чому бажане використання різних серверів, оскільки при збої наприклад основного сервера обробки тексту, було б бажано отримувати статистику на основі попередньо оброблених результатів не виконуючи операцію обробки. При використанні трьохланкової архітектури з одним сервером при його падінні чи помилці при передачі запиту, користувач не зможе взагалі працювати з системою, не те щоб, навіть отримати результати статистики. Звичайно, можна використати ще один сервер повністю зкопійований з першим, або навіть декілька, що з’єднані до балансувальника, однак при зміненні коду одного з модулів системи, прийдеться міняти увесь проект і запускати його перекомпілювання для кожного з таких серверів окремо. В контексті даного проекту такий результат не є прийнятний, кожна підсистема не повинна бути залежною від інших, а, наприклад, система прогнозування не повинна залежати від системи обробки тексту, оскільки при її збої чи помилці обробки, результат прогнозування можна отримати на основі попередньо оброблених раніше даних, оскільки динаміка за день не повинна значно змінюватися і в той же час, змінювати результати прогнозу. По-друге, через складність роботи та залежність від інструментів обробки даних, що повинні розгортатися на окремому сервері чи в хмарному середовищі, залежність одного сервера від таких модулів може значно зповільнити загальну роботу додатку в цілому, а оскільки основним критерієм роботи є швидкість, то використання такого типу архітектури не підходить, більше того, якщо всі модулі додатку будуть розташовані в одному місці, то загальна характеристика системи, щодо її

оновлюваності в майбутньому може бути значно малою, а додавання нової функціональності, як наприклад, пошуку нових деталей чи додавання нової соціальної мережі як джерела може бути тяжким, при чому при аналізі даних та отриманні даних з декількох соціальних мереж, загальний час роботи може значно впасти, що призведе до неочікуваних раніше результатів. Такі проблеми виникають при використанні трьохланкової архітектури в багатьох великих системах, такі системи ще називають монолітними. Однак, рішення для усіх цих проблем є очевидним, можна розділити кожен із модулів системи на менші, слабкозв'язані, саме такі рішення надає використання мікросервісної архітектури, що стала популярною декілька років тому.

Офіційного визначення для мікросервісів немає, проте суть полягає у тому, що мікросервіси представляють архітектурний стиль, в якому складні додатки створені як сукупність маленьких, легких, самодостатніх, незалежних, нетісно зв'язаних сервісів, кожен з яких відповідальний за конкретний процес. Такий стиль протиставляється монолітному стилю, згідно з яким додатки будуються як єдине ціле. Мікросервіси співпрацюють один з одним на основі потреби виконання певної дії. Вони «спілкуються» через API, для яких не має значення мова програмування. Цей підхід нагадує команду розробників, де кожен учасник сконцентрований на конкретній задачі. Йому дають певну відповідальність, свободу і довіру щодо виконання своєї ділянки роботи найкращим чином. Можна застосовувати мікросервісну архітектуру в побудові додатку з нуля і , можна, звичайно, розбити монолітний додаток на сервіси. Розглянемо основні переваги даного підходу розробки[17]:

1. Швидке внесення змін та розгортання. Кожен мікросервіс розгортається окремо, а отже, якщо розробник змінює щось в одному з них, то він може розгорнути ці зміни, не чіпаючи інші мікорсервіси, що можуть продовжити працювати як і раніше. При чому, зміни можуть вноситися настільки часто, наскільки це потрібно, щоб конкретна частина додатку

відповідала новим потребам бізнесу і не зачіпала весь додаток вцілому, навідміну від описаної раніше монолітної архітектури.

2. Повністю модернізовані додатки. Будь-який мікросервіс можна легко замінити чи оновити. Його можна переписати заново в межах прийнятного часу без необхідності переписувати усю систему вцілому. Таку систему легко розширити та переробити, більш того така система може завжди змінити свою “форму” на більш нову та покращену.
3. Потенційно легша система для сприйняття, підтримки та тестування. Мікросервіси, зазвичай, є невеликими за обсягом коду. Завдяки цьому, команді розробників легше зрозуміти процеси в системі, а також таку систему набагато легше підтримувати та вносити зміни, оскільки розробники сконцентровані на конкретних компонентах і впевнені, що зміна їхнього коду не зламає інші частини додатку. Більш того, таку систему набагато легше покрити автоматичними тестами. При чому тестувальникам також необхідно перевіряти маленькі частини системи, а не усю систему в цілому, не потребуючи розгортання повної системи на тестових серверах.
4. Помилка в одному мікросервісі не підірве роботу системи. Неполадки у окремому мікросервісі не повинні зламати систему. Скоріше за все, помилка в окремому сервісі не викличе злам системи.
5. Мікросервіси, написані на різних мовах програмування можуть працювати разом. Як будь-які учасники команди, мікросервіси можуть бути різними, як наприклад, система може мати мікросервіси написані на Java і на .NET, все, що повинні знати сервіси – як комунікувати між собою, при чому в основному використовується Web API на основі HTTP запитів, а також сервісної шини.

Хоча мікросервісні системи мають значні переваги над монолітними системами, ідеального рішення не існує, мікросервісні додатки мають ряд своїх недоліків, які потрібно описати. Основними недоліками є складність розробки,

оскільки система складається з менших, незалежних компонентів, підтримка роботи та обробка запитів є набагато складнішою ніж з використанням монолітного підходу. Один із мікросервісів може не відповідати на запити, що може заставити розробників писати більше коду для обробки такої ситуації. Ще одним недоліком є підтримка транзакцій для баз даних, розподілених серед багатьох мікросервісів у системі. Розгортання мікросервісів хоча і є швидшим, проте все ж, можуть бути проблеми, як наприклад, написання окремої логіки для координації. Підтримка безпеки для такого типу додатків є набагато складнішою, оскільки кожен запит повинен бути захищений спільним токеном, або використовувати один і той же механізм захисту, в монолітній архітектурі підтримка безпеки є набагато легшим процесом, оскільки додатку потрібно перевірити права користувача лише один раз на запит, оскільки усі підсистеми знаходяться в одному місці, в мікросервісній архітектурі додаток не є розподіленим, тому при використанні декількох сервісів при обробці запиту, кожен сервіс повинен передавати маркери того, що користувач має права на виконання даної операції. Хоча, мікросервісна архітектура має багато недоліків в порівняні з монолітним підходом, проте виграші, що вона надає є набагато кращими для даної системи, а, оскільки хмарні середовища широко підтримують принципи контейнеризації, то розгортання системи, що використовує мікросервісну архітектуру є гарним варіантом.

2.4 Висновки за розділом

В даному розділі було розглянуто основні підходи та методи до виконання прогнозування значень на основі історичних даних, такі як ARIMA модель, лінійна регресія а також її вдосконалення та рекурсивні нейронні мережі. Для кожного з типів було зазначено основні відмінності, переваги та недоліки, наприклад, лінійну регресію можна використовувати для прогнозування даних на коротких інтервалах до місяця, а ARIMA моделі навпаки, для прогнозування на великі інтервали, нейронні мережі можна використовувати для малих і великих

інтервалів проте вони використовують багато процесорного часу і потребують значного часу для навчання. Усі ці підходи буде порівняно у наступному розділі.

Задачу парсингу буде вирішено за допомогою операції токенізації для великих текстів і вибору необхідних ключових значень для тих соціальних мереж, які не надають зручного API для отримання даних, для інших буде використовуватися стандартний провайдер JSON даних для отримання інформації.

Оскільки дані будуть отримуватися у великих кількостях, то необхідно використовувати інструменти для їх обробки та групування для подальшого виконання прогнозування. Було порівняно різні методи, такі як використання великої кількості серверів з СУБД, MPI, Hadoop MapReduce та Apache Spark, вдалим вибором є Apache Spark і саме його буде використано у практичній частині.

Архітектура додатку грає найважливішу роль на всіх стадіях його розробки, було показано що вибір веб-сервісної архітектури, а саме мікросервісної є вдалим вибором для парсера даних з соціальних мереж, оскільки це дасть змогу швидко обробляти дані та захистить сервіс від перевантажень та виходу з ладу.

В наступному розділі буде детально описано основні сервіси додатку та їх роботу.

3 РЕАЛІЗАЦІЯ ПАРСЕРА ГЕТЕРОГЕННИХ ДЖЕРЕЛ ДАНИХ

3.1 Огляд архітектури та основних компонентів системи

Як зазначалося в попередньому розділі, для реалізації додатку було обрано мікросервісну архітектуру для реалізації парсеру, а отже, додаток повинен складатися з маленьких частинок-сервісів, кожна з яких виконує певну, покладену на неї роботу. Було вирішено, що додаток буде скдатися з семи основних частин, кожна з яких виконує різну роботу – дозволяє користувачу взаємодіяти з іншими сервісами, відповідає за авторизацію і т.д. Загальний список компонентів системи:

1. Сервіс користувацького інтерфейсу – відповідальний за усі види взаємодії користувача із системою, об’єднаний майже із усіма сервісами у системі, саме з цього сервісу починається робота усіх процесів у системі;
2. Сервіс авторизації та аутентифікації – відповідає за процес створення нового користувача, оновлення профілю та основних даних про користувача, генерує токени для доступу користувача до інших систем, також є центральним компонентом для перевірки прав користувачів на виконання різних типів дій;
3. Сервіс отримання результатів з соціальних мереж – відповідає за отримання та парсинг даних із соціальних мереж різних типів, оскільки соціальні мережі не мають стандартизованого набору команд та інтерфейсів для взаємодії, то даний сервіс реалізує провайдери та адаптери для збору даних для системи;
4. Сервіс обробки результатів із соціальних мереж – використовується сервісом отримання результатів для відповідної швидкої обробки даних;
5. Сервіс прогнозування – відповідає за виконання прогнозування майбутніх значень послідовності, що було отримано та збережено сервісом отримання результатів із соціальних мереж, для виконання прогнозування використовуються алгоритми, описані у другому розділі;

6. Сервіс моніторингу та діагностики системи – використовується для логування помилок та збоїв у роботі сервісів системи та додатку у цілому.

Детально функціональність кожного із сервісів буде розглянуто у наступних розділах, проте взаємодію сервісів між собою зображено на рис. 3.1:

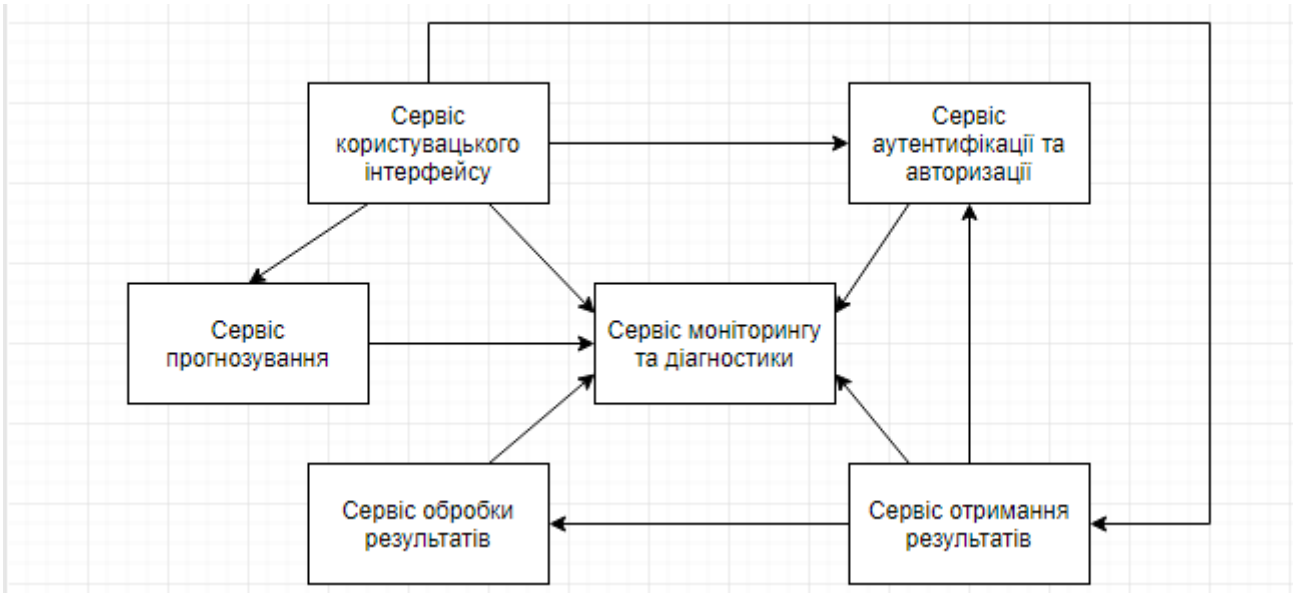

Рисунок 3.1 – Схема роботи додатку та основних компонентів

Для реалізації кожного із сервісів використовується ряд фреймворків та засобів, для реалізації серверної частини використовується фреймворк ASP.NET Core, для тих компонентів, що мають користувацький інтерфейс було вирішено використати бібліотеку React. Для забезпечення взаємодії між кожним сервісом використовується REST API з використанням протоколу HTTP, дані передаються у вигляді JSON. Деякі сервіси мають джерело збереження даних, а саме SQL бази даних, як наприклад сервіс аутентифікації та авторизації, де дані про користувача зберігаються у вигляді таблиць. Для зібрання усіх сервісів у контейнери використовується Docker. Сам додаток розміщено у хмарному середовищі у вигляді контейнера, як буде зазначено у четвертій частині диплому, з деяких причин, найкращим вибором став Google Application Engine. Деякі з сервісів використовують сторонні API для роботи, як наприклад, сервіс отримання результатів із соціальних мереж, проте усі ці сервіси використовують ті ж засоби комунікації, як і реалізований додаток.

3.2 Сервіс взаємодії користувача з системою

Сервіс взаємодії користувача з системою є чи не найбільш важливим у системі. Даний сервіс вміщує у себе основні функції передачі та прийому даних від інших сервісів через інтерфейси API та передачу JSON даних на клієнтській стороні з використанням Javascript фреймворка ReactJS. Основними кроками при взаємодії користувача з системою є:

1. Вхід користувача на веб-сайт сервісу;
2. Введення користувацького пароля та елекронної пошти в відповідній формі входу;
3. В відкритому вікні користувачу надається можливість редагувати профіль, запускати задачі з прогнозування та аналітики;

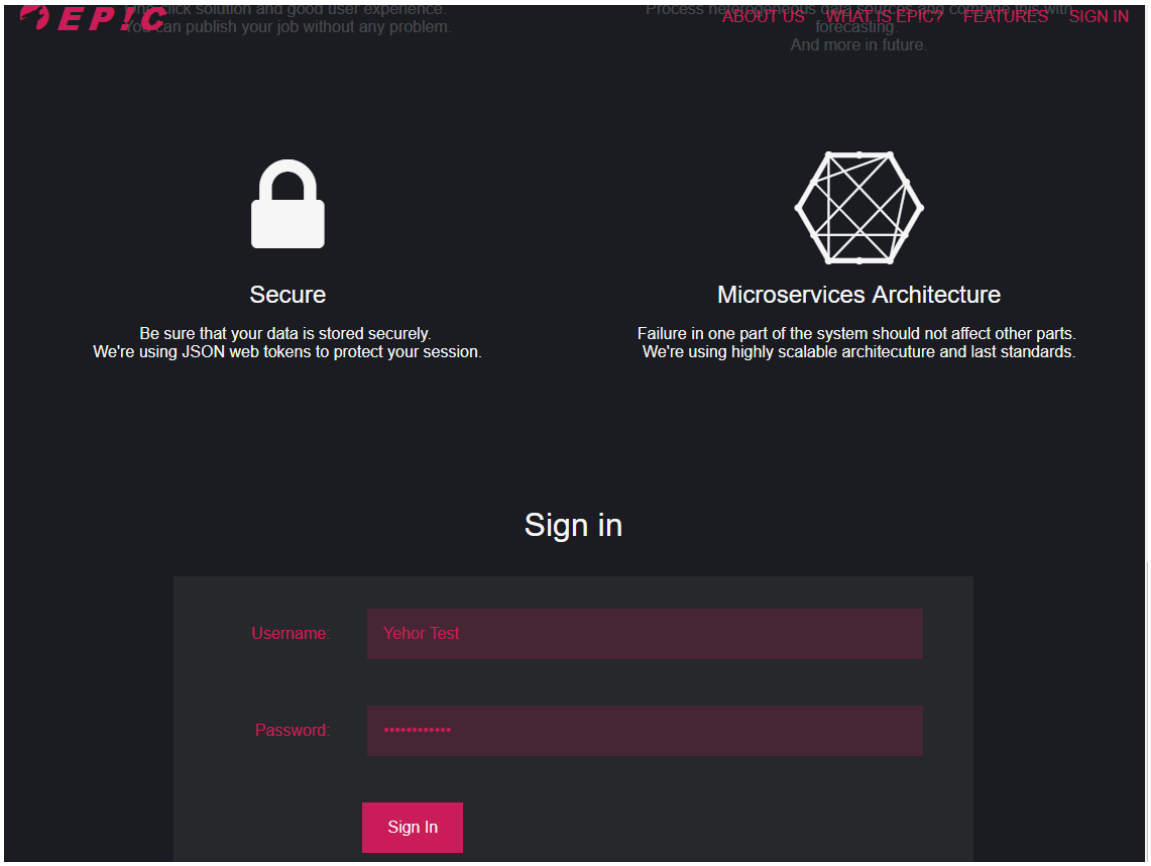

Рисунок 3.2 – Приклад форми входу на головній сторінці сайту

При вході у власний користувацький кабінет користувач може поміняти свої дані, такі як пароль, електронну пошту, логін та інше, тобто основні

стандартні поля будь-якої веб-системи. При цьому відбувається передача данних до сервісу авторизації та аутентифікації, про який буде докладніше описано в наступній секції. Користувач може запустити процес аналітики, а саме парсингу даних та статистики по фреймворкам, компаніям та спеціальностям, загалом користувач може вибрати доступні соціальні мережі з пошуку робіт такі як LinkedIn, rabota.ua, dou і інші зі списку, Також користувач може обрати необхідну спеціальність для пошуку, як наприклад “Senior Java Engineer” чи “Android Engineer” і т.д., в результаті сервіс користувацького інтерфейсу відправить запит до сервісу парсингу та обробки даних з доступних соціальних мереж, в залежності від кількості інтервалів часу, обраних користувачем та кількості обраних соціальних мереж, дані можуть бути повернуті майже миттєво чи через деякий час, при чому при довгій обробці даних система відправить лист на пошту користувача про те, що його запис було успішно оброблено.

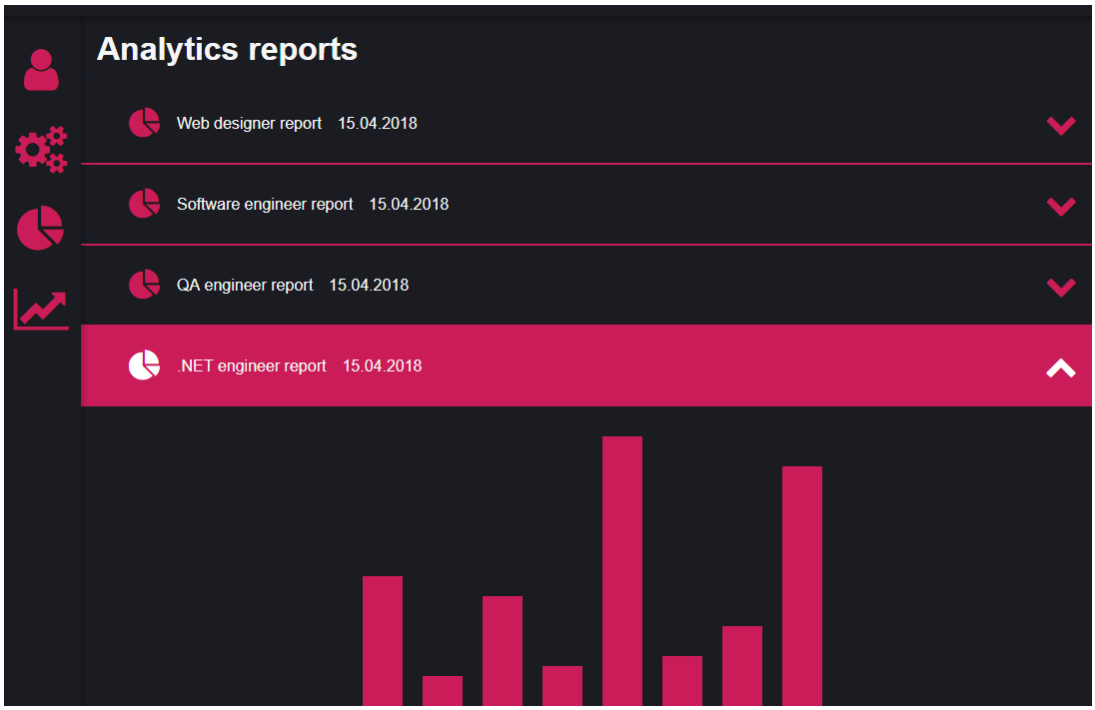

Рисунок 3.3 – Приклад користувацького інтерфейсу аналітики

Користувач також має можливість виконати прогнозування, використовуючи отримані дані з минулих даних аналітичного розбору, як, наприклад, отримати дані на тиждень чи на місяць вперед, щодо популярності тієї

чи іншої професії, використання прогнозу на рік, нажаль, дасть неточні результати через брак даних, оскільки всі алгоритми прогнозування, що використовує система є залежними від даних попередніх дослідів, а, отже, для отримання даних на рік вперед, маючи вибірку у декілька місяців, можливо, проте результат буде неточним з досить високою ймовірністю. При операціях аналізу та прогнозування користувачу буде виведено графік з інтервалами часу та кількістю співпадінь за цей період часу, як зображено на рис. 3.3.

Загалом, весь процес взаємодії користувача з системою можна зобразити на UML діаграмі послідовності таким чином, як на рис. 3.4. Потрібно зазначити, що на рисунку сервіс користувацького інтерфейсу позначено як “UIService”, сервіс авторизації – “AuthService”, сервіс аналітики – “AnalyticsService” і “ForecastingService” відповідає сервісу прогнозування відповідно.

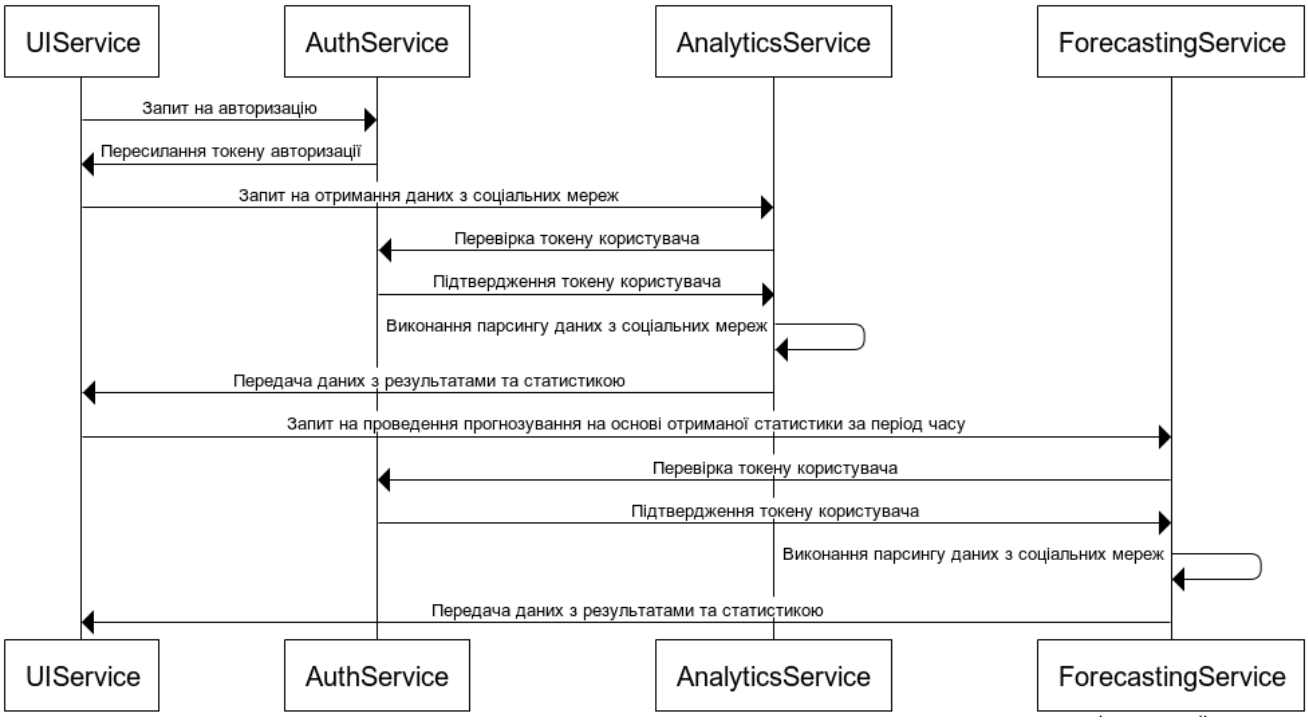

Рисунок 3.4 – UML діаграма послідовності для сервісу користувацького інтерфейсу

3.3 Сервіс авторизації та аутентифікації

Оскільки для даного додатку необхідно зберігати раніше отримані дані, а також дані для додатку, то було вирішено додати сервіс авторизації та аутентифікації, що зберігає дані про користувача, а також видає ключі для використання інших сервісів у проекті. Для підтримки процесу авторизації було обрано механізм JSON Web Tokens (JWT). JWT не є новим механізмом, проте дуже гарно підходить для мікросервісної архітектури.

Для того, щоб використовувати сервіси прогнозування та аналітики користувачеві необхідно в відповідні формі ввести свій пароль та електронну пошту, в результаті сервіс користувацького інтерфейсу відправить відповідний POST запит на генерацію JWT токену для користувача. З даним токеном користувач є авторизованим в системі, що дозволяє йому використовувати всі можливості системи. На рис. 3.5 зображено структуру типового JWT токена:

Рисунок 3.5 – Структура типового JWT токена

Кожен токен складається з трьох компонентів розділених крапкою, кожен з яких має свої функції при розпізнанні сервером. Червоним кольором на рис. 3.5 позначено хедер токену, в якому збережена інформація про алгоритм, що було використано для шифрування секретної частини токену, хедер – не що інше як JSON об’єкт, а сам хедер надалі форматується у base64 строку як показано на рисунку 3.5. Наступним компонентом є основна частина токену в якій і зберігаються дані користувача, виділена фіолетовим кольором на рисунку. В даній частині можуть зберігатися ролі користувача у системі, пошта, ім’я і т.д. Ця частина токену також форматована у вигляді об’єкта JSON і форматується у

вигляд base64 строки, тобто її можно розшифрувати на клієнті у разі необхідності і отримати необхідні дані про користувача. Найважливіша частина токену виділена синім, ця частина є гарантією того, що серверу передали достовірний токен і саме з цією частиною працює сервер авторизації для перевірки даних доступу користувача до компонентів системи. Вона формується з використанням алгоритму, зазначеного в хедері і розраховується так:

$$T = Algo(base64(header) + "." + base64(signature), secretKey) \quad (3.1)$$

Звичайно, як видно з формули сервер повинен мати секретний ключ для генерації даної частини токену, з допомогою якого він зможе розшифрувати секретну частину за алгоритмом у хедері токену та перевірити чи є правильними отримані хедер та основна частина токену.

Однак, це не єдина функціональність для даного сервісу, оскільки він відповідає не лише за генерацію і перевірку токену користувача, а й збереження та оновлення даних користувача. Користувач має можливість змінювати пароль у системі, інші дані, а також видаляти свій акаунт із системи, оновлювати основну інформацію про себе, а також переглядати профілі інших користувачів, якщо це необхідно, звичайно, якщо у користувача є відповідні права, для цього на даному сервісі реалізовано такі ендпоінти для використання іншими компонентами.

Таблиця 3.1 – Основні ендпоінти сервісу авторизації та аутентифікації

Метод	URL	Дані	Відповідь
POST	/auth/signin	Пароль та електронна пошта	Згенерований токен
GET	/auth/verify	Токен користувача	Результат перевірки
GET	/user?id	Id користувача	Дані про користувача
POST	/user/	Інформація про користувача	Результат реєстрації
PUT	/user/	Оновлена інформація	Результат оновлення
DELETE	/user/	Id користувача	Результат видалення

Загалом, весь процес взаємодії інших сервісів з сервісом автентифікації та авторизації можна зобразити на UML діаграмі послідовності, де “AuthService” –

сервіс авторизації та аутентифікації, “UIService” – сервіс користувацького інтерфейсу, а “ForecastingService” – сервіс прогнозування:

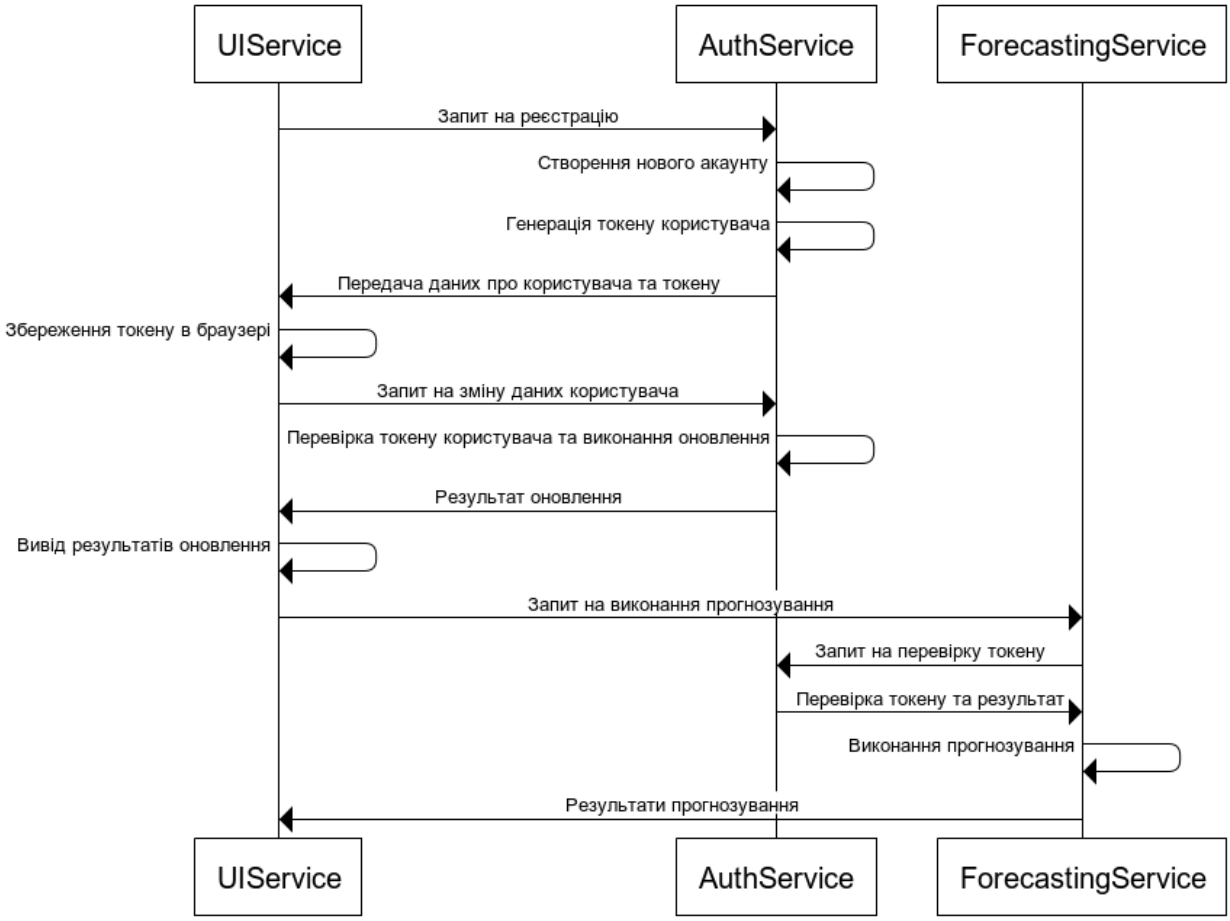

Рисунок 3.6 – UML діаграма послідовності для сервісу авторизації та аутентифікації

3.4 Сервіс отримання даних з соціальних мереж та парсингу

Сервіс отримання даних та обробки є ключовим для системи, оскільки саме в цьому сервісі виконується збір інформації з соціальних мереж і видається результат згідно запиту користувача. Даний сервіс є найбільним і найскладнішим в імплементації, оскільки соціальні мережі не мають єдиного стандарту доступу до даних, а деякі і зовсім не надають API для роботи з даними для отримання вакансій по різним фреймворкам та компонентам. Загалом, через такі складнощі, а також через гетерогенність і постійну оновлюваність даних було обрано чотири основні соціальні мережі з пошуку роботи та викладення вакансій користувачів, а

саме Dou, Stackoverflow, Jooble та LinkedIn. При чому кожен із даних сервісів має свої способи для отримання даних. LinkedIn, наприклад, не надає доступу до вакансій компаній, проте надає доступ до інформації кандидатів, технологій, що вони використовують та компаній, в яких вони працюють, Jooble надає доступ до свого API в якому можна отримати кількість вакансій за ключовим ім'ям:

```
{
    "keywords": "account manager",
    "location": "London",
    "radius": "50",
    "salary": "200000",
    "page": "1"
}
```

Рис. 3.7 – Форматований запит на API соціальної мережі

В результаті такого запиту буде зформовано результат у вигляді кількості вакансій і датах, коли вакансію було в останнє змінено, на рис. 3.8 зображено приклад відповіді на запит, в якому показано кількість викладених вакансій по заданій професії, в даному випадку "Account Manager", а також заробітня плата та інша необхідна інформація, для даної задачі основними даними є кількість вакансій загалом та час оновлення, для того, щоб згенерувати графік та використати отримані дані для прогнозування.


```
{
    "totalCount":651,
    "jobs": [
        "job": {
            "title": "Account manager",
            "location": "London",
            "snippet": "... are looking for account manager in London, greate opportunity to ...",
            "salary": "200000 $",
            "source": "barclays.co.uk",
"type": "Any",
            "link": "https://us.jooble.org/away/...",
            "company": "Jooble",
            "id": 8240559805230395300,
            "updated": "2016-05-26T10:51:51.4720673+03:00"
        },
        "job": ...
    ]
}
```


Рисунок 3.8 – Приклад відповіді на запит до соціальної мережі пошуку роботи

Інші сервіси, такі як Stackoverflow та dou.ua не надають публічних API для доступу до своїх даних, тому сервіс виконує запит на такі ресурси і виконує токенізацію компонентів HTML даних, виділяючи необхідні дані, приклад доступних даних зі Stackoverflow:

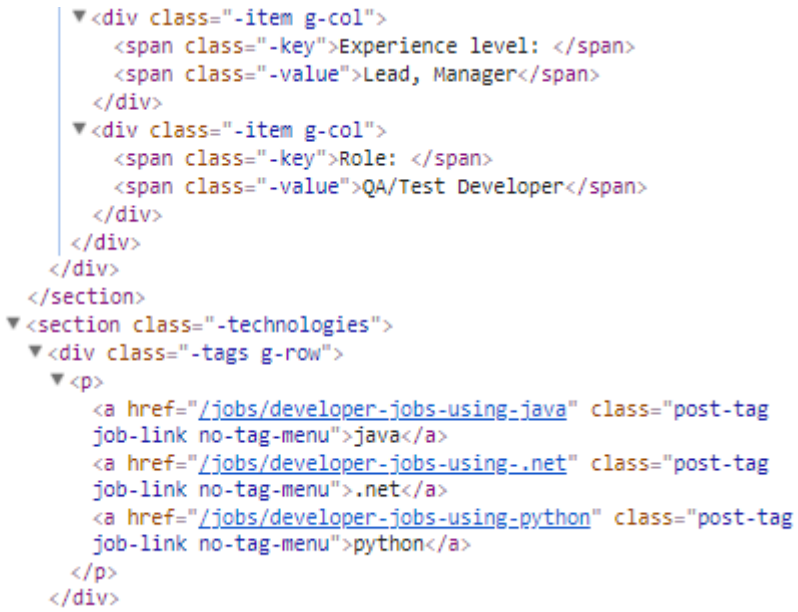

Рисунок 3.9 – Приклад даних з Stackoverflow

Як видно з зображення, кожна вакансія містить у собі необхідні технології для кандидата, рівень його досвіду, необхідний для вакансії, а також назва вакансії, час, заробітню платню та інше саме з цього можна отримати дані аналітики за певні періоди часу.

Сервіс отримання даних надає можливості користувачеві обирати тип запиту – вакансії, технології, рівень досвіду кандидата по різним періодам часу, тиждень, день, години, роки і т.д., формуючи звіт і відправляючи відповідь на сервіс користувацького інтерфейсу, потрібно зазначити, що для обробки даних за один день, місяць, годину або тиждень використовується обробка на стороні сервера одразу у пам’яті, а результат повертається за долю часу, проте для обробки даних за роки потрібен час, тому користувачу одразу повертається повідомлення про те, що його запит було додано у чергу обробки, а повідомлення про кінець обробки буде надіслано листом на пошту користувача, якщо ж у

користувача немає електонної пошти, зареєстрованої в додатку, то він може зазначити її одразу чи отримати результати після входу наступного разу через деякий період часу. Обробкою таких даних займається сервіс обробки даних, що відправляє запит на GAE, де розташовано Spark, який і оброблює необхідні дані з високою швидкістю, а сам сервіс обробки перевіряє з вказаним на сервері інтервалом результати обробки даних та збирає отримані дані, оскільки Spark не нотифікує про те, що дані було оброблено. Після цього отримані дані зберігаються в базі даних у вигляді звіту по конкретному запиту асоційованому з користувачем. Надалі, коли користувач зайде у систему, то зможе продивитися результати обробки даних по його запиту.

Приклад отриманих даних після оброки та парсингу з соціальних мереж у вигляді JSON, зображеного на рис. 3.10. Як видно, користувач сервісу виконав пошук даних за позицією “Account Manager” на сервісі Stackoverflow за період від 15.03.2018 до 21.03.2018, в результаті чого було згенеровано результат із послідовності дат і кількості вакансій за цю дату.


```
{
    "requestId": "434bca8a-4efb-4875-a213-136b6c049aad",
    "userId": "0f142feb-3ab8-4103-9e3f-1df68483ae15",
    "query": {
        "service": "Stackoverflow",
        "startDate": "15-03-2018",
        "endDate": "21-03-2018",
        "type": "job",
        "name": "Account manager"
    },
    "results": {
        "15-03-2018": 20,
        "16-03-2018": 21,
        "17-03-2018": 21,
        "18-03-2018": 15,
        ...
        "21-03-2018": 11
    }
}
```


Рисунок 3.10 – Приклад відповіді на запит користувача сервісом у форматі JSON

Для того, аби забезпечити такого роду взаємодію між користувачем та відповідними сервісами було реалізовано API методи для отримання даних за

певним запитом на отримання статистика, отримання всіх запитів на отримання статистики і т.д., для сервісу обробки даних було реалізовано такі ендпоінти.

Таблиця 3.2 – API ендпоінти для сервісу отримання даних

Метод	URL	Дані	Відповідь
POST	/analytics/	Необхідний сервіс, інтервал часу, вакансія чи фреймворк	Отриманий результат аналітики
GET	/analytics?id	Id звіту	Отриманий звіт
GET	/analytics/	-	Усі звіти, що було отримано користувачем раніше
DELETE	/analytics	Id звіту	Результат видалення відповідного звіту

Для того, щоб виконувати обробку результатів із сервісу отримання даних, було реалізовано API сервісу обробки, що приймає необхідні дані та оброблює їх відповідним чином, потрібно зазначити, що до сервісу обробки може виконувати запити лише сервіс аналітики. Основні ендпоінти сервісу обробки зазначено у таблиці 3.3.

Таблиця 3.3 – API ендпоінти для сервісу обробки

Метод	URL	Дані	Відповідь
POST	/processing/	Дані від сервісу ортимання даних	Оброблені дані

Загалом, весь процес взаємодії інших сервісів з сервісами обробки та аналітики можна зобразити на UML діаграмі послідовності, де “AuthService” – сервіс авторизації та аутентифікації, “UIService” – сервіс користувацького інтерфейсу, а “AnalyticsSerivce” – сервіс аналітики, “ProcessingService” – сервіс обробки даних:

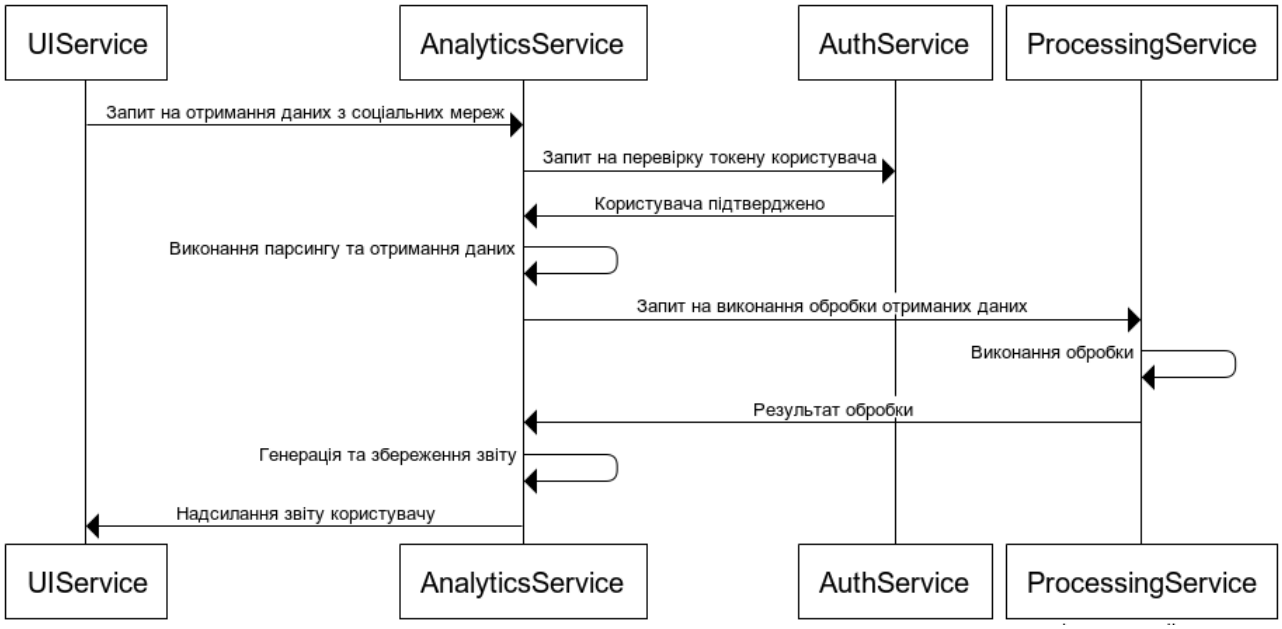

Рисунок 3.11 – Процес взаємодії сервісів аналітики та обробки

3.5 Сервіс прогнозування

Сервіс прогнозування використовується для виконання прогнозів, маючи наявні дані за попередні інтервали часу. Основні алгоритми, що використовуються для прогнозування описані в розділі 2.1, а імплементовано ці алгоритми в сервісі прогнозування. Користувач має можливість обрати необхідний алгоритм, якщо алгоритм не буде обрано, то буде використано усі, а результати будуть надані по всім типам алгоритмів. Потрібно зазначити, що для того, аби використовувати сервіс прогнозування, користувач має виконати хоча б один запит до сервісу отримання результатів з соціальних мереж, оскільки наявні алгоритми потребують попереднього набору даних. Прикладом запиту до сервісу прогнозування є вказаний JSON на рис. 3.12:

```
{
    "algorithm": "arima",
    "pointsNumber": 1,
    "reportId": "efc462bc-4627-4721-befd-b472eecc8c81",
    "results": {
        "15-03-2018": 20,
        "16-03-2018": 21,
        "17-03-2018": 21,
        "18-03-2018": 15,
        ...
        "21-03-2018": 11
    }
}
```


Рисунок 3.12 – Приклад запиту на сервіс аналітики

Як видно із рисунку на сервіс прогнозування було відправлено запит для виконання прогнозування з використанням алгоритму ARIMA для прогнозування одного значення, базуючись на звіті сервісу аналітики із вказаним id, з отриманими результатами звіту аналітики. В результаті буде використано ARIMA прогнозування, створено відповідний звіт для прогнозування, а результат буде повернуто користувачу у вигляді JSON на рис. 3.13:


```
{
    "reportId": "a792b523-1abe-4df2-bc05-939d3e3a3992",
    "basedOnReportId": "efc462bc-4627-4721-befd-b472eecc8c81",
    "algorithm": "arima",
    "forecasting": {
        "22-03-2018": 10
    }
}
```


Рис. 3.13 – Приклад результату сервісу прогнозування

Усі звіти для кожного користувача зберігаються у базі даних, кожен користувач може отримати доступ до виконаних звітів, переглядати їх, створювати нові. Для того, щоб забезпечувати такі можливості було реалізовано таку функціональність за такими ендпоінтами як наведено в таблиці 3.4.

Таблиця 3.4 – Список API ендпоінтів для сервісу прогнозування

Метод	URL	Дані	Відповідь
POST	/forecasting/	Алгоритм та звіт аналітики	Отримані дані з прогнозування
GET	/forecasting/	-	Усі прогнози виконані користувачем
GET	/forecasting?id	Id прогнозу	Дані за звітом прогнозу
DELETE	/forecasting/	Id прогнозу	Результат видалення прогнозу

Для сервісу прогнозування UML діаграма послідовності схожа на ту, що наведено на рис. 3.6, тому у даному розділі її наведено не буде.

3.6 Сервіс моніторингу та стану додатку

Сервіс моніторингу та стану додатку не використовується користувачами додатку напряму, навпаки, сервіси самі відправляють інформацію про помилки та виключення на програмному рівні. Типовою помилкою додатку може бути недоступність конкретного сервісу внаслідок збою, чи низької швидкості підключення до мережі, внаслідок чого сервіс не зміг виконати поставлену на нього задачу. Такий сервіс є корисний для діагностики додатку уцілому, для визначення основних проблем чи нових викликів. Наприклад, представимо, що API однієї із соціальних мереж було змінено та оновлено без можливості використання старого API, в результаті сервіс отримання даних з соціальних мереж не зможе оброблювати дані з соціальної мережі, такі ситуації потрібно вирішувати як можливо скоріше, тому отримання швидкого способу для знаходження помилок є критичним. Ще одним прикладом ситуації, в якій може бути необхідним використання такого сервісу – виконання вимірів часу обробки даних конкретним алгоритмом чи сервісом, в результаті чого можна виявити неоптимізовані частини додатку та застосувати необхідіні засоби для покращення часу роботи та рефакторінг. Оскільки сервіс моніторингу та стану додатку не має особливо тяжких завдань, то і його API не є занадто складним, таблиця 3.5.

Таблиця 3.5 – API ендпоінти для сервісу моніторингу та стану додатку

Метод	URL	Дані	Відповідь
POST	/monitoring/	Дані про стан сервісу	–
GET	/monitoring/	Фільтр	Усі дані моніторингу

В основному сервіси виконують запити для запису логів моніторингу свого стану, використовуючи POST метод. Користувач з відповідними прівілегіями може продивитися логи стану з використанням сервісу користувацького інтерфейсу.

3.7 Висновки за розділом

В даному розділі було описано основні компоненти додатку-парсеру, було продемонстровано основні принципи роботи та зв’язок між компонентами системи. Як було зазначено, даний парсер-сервіс реалізовано за допомогою мікросервісної архітектури, що дозволило розділити додаток на маленькі частини, де кожна виконує лише одну задачу, таке розмежування допомагає зконцентрувати логіку на конкретному сервісі і виокремити логіку залежності одного сервіса від іншого.

Загалом додаток складається із семи основних сервісів – сервісу користувацького інтерфейсу через який виконується взаємодія користувача із системою, а саме усіма сервісами, наявними у додатку, сервісу аутентифікації та ауторизації, який відповідає за реєстрацію, вхід та основні операції користувача із власним профілем, сервісу отримання та парсингу даних з соціальних мереж, який є ключовим і дозволяє отримувати дані із соціальних мереж. Сервіс обробки даних виконує необхідну обробку отриманої інформації для її подальшого збереження і використання в наступних запитах чи прогнозуванні. Сервіс прогнозування, на основі отриманих аналітичних даних, виконує прогнозування на подальші періоди часу, зазначені користувачем. Сервіс моніторингу

використовується усіма сервісами системи для логування помилок та трекінгу часу роботи окремих компонентів.

В наступному розділі буде розглянуто реалізацію стартап-проекту, прораховано бюджет проекту, необхідні затрати та доцільність виходу такого проекту на ринок.

4 РЕАЛІЗАЦІЯ СТАРТАП-ПРОЕКТУ “WEB ANALYTICS PARSER”

4.1 Опис ідеї та технологічний аудит стартап-проекту

У даному розділі описано економічне обґрунтування реалізації стартап-проекту на тему «Парсер гетерогенних джерел даних з можливістю прогнозування». За допомогою даної програми-парсера можна отримувати дані з різних соціальних мереж пошуку роботи та, використовуючи наявні алгоритми прогнозування, виконувати прогноз раніше отриманих даних на майбутні періоди часу. В результаті система повинна зберігати дані про користувачів, що користуються системою, зберігати результати обробки даних з соціальних мереж, виконувати прогнозування та зберігати звіти з прогнозів.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вимоги для користувача
Ідея полягає в тому, щоб створити систему, завдякі якій можна швидко отримувати дані з різних гетерогенних джерел, таких як соціальні мережі, для використання в різних галузях.	1. Вирішення задачі розподіленної обробки даних	Користувачу необхідно лише зайти в систему, система повинна сама вирішувати на яких доступних серверах запускатися та обробляти дані.
	2. Вирішення задачі пошуку та збирання необхідних даних	В користувача буде можливість швидко вирішувати задачу пошуку даних за заданими параметрами.

Як видно з таблиці 4.1 основна ідея – створення системи, що здатна швидко і точно отримувати та аналізувати дані з гетерогенних джерел, на прикладі соціальних мереж, а основні проблеми, що вирішує система – виконання розподіленої обробки даних, тобто дані оброблюються не на одному сервері, а на декількох, а також збирання та обробка необхідних даних за запитом користувача, все, що потрібно від користувача – можливість доступу до системи.

Для успішної реалізації проекту необхідно визначити його сильні та слабкі сторони, для цього проводиться порівняльний аналіз показників проекту (слабкі, нейтральні та сильні) з аналогічними проектами-конкурентами. Для цього визначаються техніко-економічні характеристики ідеї проекту, а також наявні конкуренти та їх аналогічні характеристики, в результаті проводиться порівняльній аналіз, на основі якого визначається конкурентноспроможність.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних харистик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент1	Конкурент2	Конкурент3			
1.	Форма виконання	Веб сервіс	Веб сервіс	Веб сервіс	Програма		+	
2.	Собівартість	Низька	Висока	Висока	Низька			+
3.	Наявність інтернету	Треба	Треба	Треба	Треба		+	
4.	Кросплатформенність	Так	Так	Так	Ні		+	
5.	Підтримка декількох джерел	Так	Так	Ні	Ні			+

Як видно з таблиці 4.2 було визначено перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного сервісу, що є підґрунтям для формування його конкурентоспроможності. Сильною стороною проекту є його низька собівартість, оскільки основні технології, що використовуються в проекті є безкоштовними і з відкритим доступом, що впливає на ціну послуг для користувача, також даний сервіс використовує та оброблює дані з декількох гетерогенних джерел, чого немає у більшості конкурентів, що сфокусовані на конкретних соціальних мережах, інші характеристики є нейтральними, оскільки не мають явних переваг, або збігаються з

характеристиками конкурентів. В результаті отриманих даних, можна зробити висновок, що проект є конкурентоспроможним.

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових: за якою технологією буде виготовлено товар згідно ідеї проекту, чи існують такі технології, чи їх потрібно розробити/доробити, чи доступні такі технології авторам проекту. Отримані дані наведено в таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність проекту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1.	Швидкий парсер даних з гетерогенних джерел. Реалізація серверної частини	Amazon Web Services	Наявна	Платна, недоступна
		Google App Engine	Наявна	Безкоштовна (до певних розмірів)
		Microsoft Azure	Наявна	Платна, недоступна
Обрана технологія реалізації серверної частини проету – Google App Engine, оскільки вона є безкоштовною та доступною.				

Як видно з таблиці 4.3 основною частиною проекту є його серверна частина, на яку і сконцентровано більшість затрат. Для реалізації даної частини сервісу можна використати надані в таблиці 4.3 технології, проте використання Google App Engine має найбільшу перевагу – вона є бескоштовною для даного сервісу, враховуючи його розміри.

4.2 Аналіз ринкових можливостей

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та

пропозицій проектів-конкурентів. Спочатку проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 5.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	3
2.	Загальний обсяг продаж, грн/ум.од	25000 грн/ум. од
3.	Динаміка ринку (якісна оцінка)	Стагнує
4.	Наявність обмежень для входу (вказати характер обмежень)	Немає
5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі (або по ринку), %	R = 25%

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект. Як видно з таблиці 4.4 ринок у даній галузі є стагнуючим, оскільки нових гравців у ринку аналізу даних з гетерогенних джерел немає, також даний сервіс не потребує специфічних вимог до стандартизації чи сертифікації, а середньою нормою рентабельності по галузі є 25 відсотків і ринок є привабливим для входження.

Надалі потрібно визначити основні групи клієнтів, їх вимоги до створюваного продукту, потреби ринку в даній області.

Як видно з таблиці 4.5 основною потребою ринку є використання засобів для аналізу даних з гетерогенних джеред, де цільовою аудиторією можуть бути спеціальні агенства зі збору даних, рекламні агенства, що займаються дослідженням даних. Основними вимогами, котрі повинний задовольняти

реалізований сервіс є точний результат, швидке виконання та інтуітивний інтерфейс.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Необхідно програмне забезпечення, що може швидко отримувати та аналізувати дані з гетерогенних джерел за певними критеріями	Потенційними цільовими групами є агенства зі збору даних, дослідницькі інститути, рекламні агенства	Цільова група займається дослідженнями	Швидке рішення, інтуітивний інтерфейс, точний результат.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають. В таблиці 4.6 наведено основні ринкові загрози - події, настання яких може несприятливо вплинути на підприємство.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загроз	Можлива реакція компанії
1.	Конкуренція	Вихід на ринок великої компанії	1. Вихід з ринку 2. Обрати нову цільову аудиторію і зосередитися на ній 3. Об’єднання з компанією-конкурентом
2.	Економічний	Подорожчення вартості та обслуговування сервісів, необхідних для роботи системи	1. Оптимізація програмного продукту, для можливості його запуску на більш бюджетних пристроях.

№ п/п	Фактор	Зміст загроз	Можлива реакція компанії
3.	Зміна потреб користувачів	Користувачам необхідне ПЗ з іншим функціоналом	1. Передбачити можливість додавання нових функцій
4.	Законодавчий	Зміни в законодавстві стосовно обробки персональних послуг користувачів.	1. Впровадження контроль за збереженням персональних даних користувачів
5.	Шкідливе ПЗ	Атака хакерів на сервери	1. Використання засобів захисту проти подібних атак

В таблиці 4.6 були наведені фактори загроз та способи зменшення ризиків. Найбільшою загрозою є конкуренція. Для боротьби з конкуренцією нам необхідно передбачити найкращий набір функціоналу. Також необхідно передбачити можливість додавання нового функціоналу, або зробити фокус на більш вузькій цільовій аудиторій, надаючи їм більше уваги. Також, при виході на ринок дуже потужного конкурента, можна розглянути можливість об’єднання або поглинання з метою збереження вкладених в даний проект коштів.

Надалі необхідно зазначити основні фактори можливостей – фактори при яких даний продукт матиме більший прибуток чи явну перевагу над конкурентами.

В таблиці 4.7 розглянуто фактори можливостей. Найбільш цікавим для нас звичайно є зниження довіри до конкурентів, адже це автоматично піднімає наші позиції серед конкурентів. Також вагомим є фактор збільшення можливостей покупців, адже це збільшить кількість потенційних клієнтів, які захочуть використати сервіс, що в свою чергу підніме рівень прибутків.

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Науково-технічний	Тенденція до випуску покращеного спеціалізованого обладнання та розробка більш ефективних алгоритмів	1. Адаптація існуючого рішення і алгоритмів під нову технологію.
2.	Попит	Більш широке розповсюдження технології рекомендаційних систем	1. Постійна підтримка продукту.
3.	Зростання можливостей потенційних покупців	Збільшення доходу в певній галузі	1. Запропонування своїх послуг компаніям в цій галузі
4.	Зниження довіри до конкурента	У конкурента нещодавно була знайдена помилка, витік конфіденційної інформації	1. При виході на ринок звертати увагу покупців на безпеку нашого ПЗ
5.	Популярність нового джерела даних	Вихід нової соціальної мережі, що стала популярною, як результат – необхідність в аналізі даних з такої мережі	1. Використання архітектури, що легко розширюється.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 4.8).

В таблиці 4.8 наведено аналіз конкуренції на ринку. Визначено, що най сервіс працює в середовищі нецінової, товарно-родової конкуренції, адже конкуренти частково повторюють функціонал сервісу, тож конкуренція ведеться за рахунок якості надання послуг. Конкуренція відбувається не лише в середині країни, а й на міжнародному ринку надання послуг.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – нецінова конкуренція.	Існує 3 фірми- конкуренти на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
2. За рівнем конкурентної боротьби – міжнародний	Всі компанії з різних країн	Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок
3. За галузевою ознакою – внутрішньогалузева	Конкуренти мають ПЗ, яке використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях
4. Конкуренція за видами товарів – товарно-видова	Види товарів є однаковими, а саме – програмне забезпечення	Створити ПЗ, враховуючи недоліки конкурентів
5. За характером конкурентних переваг – нецінова	Вдосконалення технології створення ПЗ	Використання менш дорогих технологій для розробки, ніж конкуренти
6. За інтенсивністю – не марочна	Бренди відсутні	–

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі. М. Портер вирізняє п‘ять основних факторів, що впливають на привабливість вибору ринку з огляду на характер конкуренції - конкурент, що вже є в галузі (загалом - три конкуренти); потенційні конкуренти; наявність

товарів-замінників; постачальники, що конкурують за ринкову владу; споживачі. Результати наведено в таблиці 4.9.

Таблиця 4.9 – Аналіз конкуренції в галузі за М.Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальн-ики	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження на ринок	Визначити фактори сили постача-льників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки	На ринку є три основні конкуренти, найбільш схожим конкуре-нтом є конкурент1	Так можливості є, оскільки наведене рішення має гарний баланс ціна/якість	Постача-льники відсутні	Важливим для користу-вача є точність та швидкість роботи ПЗ	Товари-замінники можуть використовувати швидшу технологію та зменшити собівартість товару.

В таблиці 4.9 було досліджено аналіз конкуренції в галузі за М.Портером. Було визначено конкурента, чий продукт найбільш подібний до наданого. Також було визначено та обґрунтовано можливості виходу сервісу на ринок, наведено його основні переваги та фактори сили споживачів. Також було розглянуто фактори загрози з боку конкурентів.

Тепер необхідно визначити основні фактори конкурентоспроможності даного сервісу, що дозволить виділити основні цілі при розробці сервісу та його вдосконаленні, завдяки цьому можна буде очікувати кращий старт роботи та отримання прибутків. Результати наведено у таблиці 4.10.

Таблиця 4.10 – Обгрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обгрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Веб-сервіс	Користувач не повинен розгортати дану програму у себе, повинна забезпечуватись робота у режимі 24/7, це дозволить користувачам використовувати дане ПЗ з будь-якої точки світу.
2.	Простота інтерфейсу	Користувач повинен розуміти, що потрібно зробити, щоб запустити аналіз без сторонньої допомоги та документації.
3.	Швидкість роботи	Користувач не повинен чекати результатів обробки в днях, програма повинна оброблювати результати швидко.
4.	Точність результатів	Повинна забезпечуватися гарантія точності, на яку розраховує користувач.

Як видно з таблиці 4.10, орієнтованою архітектурою сервісу є веб-сервісна, оскільки це дасть змогу користувачу не розгортати чи встановлювати сервіс на власному комп’ютері чи серверах, користувач не повинен читати документацію, щоб використовувати основні функції системи, все повинно бути інтуітивним, ціллю сервісу є його швидкість иа точність, тому дані характеристики є критичними.

Тепер проведемо порівняльний аналіз сильних та слабких сторін в порівнянні із сервісами конкурентів, це дасть змогу зрозуміти, що потрібно вдосконалити чи переробити, оскільки проектування відбувається з врахуванням можливих вдосконалень, то додавання нового функціоналу не повинне бути проблемою. Результати наведено в таблиці 4.11.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашим підприємством						
			-3	-2	-1	0	1	2	3
1.	Веб-сервіс	10				+			
2.	Простота інтерфейсу	10		+					
3.	Швидкість роботи	15			+				
4.	Точність результатів	20				+			

Як видно з таблиці 4.11 основними характеристиками є швидкість роботи та точність результатів, тому їм надано найбільшу кількість балів, при чому даний сервіс не програє конкурентам по точності і має перевагу в швидкості роботи, це пов’язане з використанням фреймворка Spark, також сервіс є простішим у використанні, оскільки більшість конкурентів також надають свої послуги в якості веб-сервісу, то значної переваги даного сервісу в даному факторі немає.

В результаті наведених даних потрібно зформувати основні характеристики наведеного сервісу, а саме його сильні та слабкі сторони, включно з можливостями та загрозами. Результати наведено в таблиці 4.12.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: простий інтерфейс користувача, швидкість роботи.	**Слабкі сторони**: Покращення швидкості роботи є критичним чинником успіху.
Можливості: Тенденція до випуску покращеного спеціалізованого обладнання та розробка більш ефективних алгоритмів, більш широке розповсюдження технології рекомендаційних систем, збільшення доходу в певній галузі, у конкурента була знайдена помилка, витік конфіденційної інформації, вихід нової соціальної мережі, аналіз даних з такої мережі.	**Загрози:** Вихід на ринок великої компанії-конкурента, подорожчення вартості та обслуговування сервісів, необхідних для роботи системи, користувачам необхідне ПЗ з іншим функціоналом, зміни в законодавстві стосовно обробки персональних послуг користувачів, атака хакерів на сервери.

Як видно з таблиці 4.12 сильними сторонами проекту є простий інтерфейс та швидкість роботи програми, що в одночас завжди буде слабкою стороною, оскільки конкурент завжди може випустити в ринок продукт, що буде працювати швидше, що і є основною загрозою, тому постійне вдосконалення швидкості роботи є пріорітетом. Орієнтованою можливістю сервісу є його ціна використання, що є нижчою ніж у конкурентів.

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Результати зазначені в таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтарнатива (орієнтований комплекс заходів)	Ймовірність отримання ресурсів	Строки реалізації
1.	Використання Spark як основного фре-ймворка для обробки даних, використання ресурсів GAE для розгортання та роботи веб-сервісу, використання відомих методів аналізу тексту	80%	Рік
2.	Написання власного фреймворка обробки даних, розгортання веб-сервісу на власних серверах та підтримка їх роботи у режимі 24/7, розробка власного алгоритму аналізу тексту	20%	Два роки

З означених альтернатив оберемо першу, оскільки ймовірність отримання ресурсів для використання вже існуючих технологій є значно вищою, ніж при написанні власних фреймворків та рішень, оскільки строки реалізації є коротшими, а потенційна ймовірність помилки роботи є мінімальною.

4.3 Розробка ринкової стратегії проекту

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи	Готовність споживачів сприйняти продукт	Орієнто-ваний попит в межах цільової групи	Інтенси-вність конкуренції в сегменті	Простота входу у сегмент
1.	Університетські дослідження	Час виконання не є критичним у даному випадку, хоча це і є основною характеристикою.	Дослідження в області великих даних проводяться постійно.	Невисока, існує лише три конкуренти	Для університетів будуть створені безкоштовні ліцензії з скороченним функціоналом, однак вони зможуть придбати повну версію, якщо продукт сподобався
2.	Дослідницькі центри	Швидка робота відношення ціна/ якість є досить привабливими для даної групи	Дослідження в області великих даних проводяться постійно		Маючи перевагу у простоті інтерфейсу та швидкості – вийти у ринок повинно бути нескладно
3.	Рекламні агенції	Відношення ціна/якість є досить привабливими для даної групи	Дані компанії завжди заохочені в розробці передових технологій та використанні їх в своїх продуктах		Маючи перевагу у простоті інтерфейсу та швидкості – вийти у ринок повинно бути нескладно
Які цільові групи обрано: обираємо дослідницькі, рекламні агенції					

Як видно з таблиці 4.14 основними цільовими групами потенційних споживачів є дослідницькі центри, що можуть виконувати аналіз даних на замовлення, а також рекламні агенції, що можуть розташовувати рекламу на сайтах соціальних мереж в залежності від контексту та показників, отриманих з використанням створеного сервісу.

Наступним кроком є вибір базової стратегії розвидку, відповідні дані наведено в таблиці 4.15.

Таблиця 4.15 – Визначення базової стратегії розвидку

№ п/п	Обрана альтернатива розвидку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно обраної альтернативи	Базова стратегія розвидку
1.	Використання Spark як основного фреймворка для обробки даних, використання ресурсів GAE для розгортання та роботи веб-сервісу, використання відомих методів аналізу тексту	Ринкове позиціювання	Простота інтерфейсу, покращення швидкості роботи	Диференціації

В таблиці 4.15 наведено основну стратегію розвидку. Було обрано альтернативу розвидку з використанням Spark та GAE, оскільки в ній зберігається відношення ціна/якість продукту, базовою стратегією розвидку було обрано стратегію диференціації.

Тепер визначимо основні характеристики сервісу з точки зору конкуренції та оберемо стратегію конкуретної поведінки, в результаті чого зформуємо відповідну таблицю 4.16.

З результатів таблиці видно, що сервіс не є першопрохідцем на ринку, основною орієнтацією є забирання споживачів у конкурентів. Як результат, було обрано стратегію зайняття конкурентної ніші.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи буде проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати в існуючих конкурентів?	Чи буде компанія копіювати основні характеристики конкурента, і які?	Стратегія конкурентної поведінки
1.	Ні	Забирання споживачів у конкурентів.	Буде, а саме: подача ПЗ як веб-сервісу.	Зайняття конкурентної ніші.

На основі вимог споживачів з обраних сегментів до сервісу та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект, таблиця 4.17.

В таблиці 4.17 описано старатегію позиціонування даного стартап-проекту. Описано основні вимоги цільової аудиторії до товару, а саме простоту у користуванні та точність рекомендацій. Визначено базову стратегію розвитку (диференціація). Сформовано перелік основних позицій конкурентоспроможності проекту.

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвидку	Ключові конкурентоспроможні позиції власного стартап-проетку	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Простота інтерфейсу, швидкість роботи	Диференціації	Простота користувацького інтерфейсу, що дозволяє пришвидшити роботу праці-вників.	Швидкість, про-стота, безпека, точність.

4.4 Розробка маркетингової програми

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару, основними потребами якого є швидкість роботи, спрощений інтерфейс та точність результатів.

Як видно з таблиці 4.18 основними потребами даного сервісу від користувачів було повністю виконано. Додаток надає швидкий доступ до даних та обробку, реалізовано спрощений інтерфейс користувача та здійснюється точна обробка результатів прогнозування. В результаті чого, сервіс може забрати ймовірних клієнтів конкурентів. Як видно з даної таблиці, з використанням Spark для обробки великих даних система отримує значне прискорення в роботі, а основною ціллю для вдалого продукту є інтуїтивний користувацький інтерфейс та точна обробка результатів.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Швидкість роботи	Обробка з використанням алгоритмів аналізу та Spark для обробки даних значно спрощує роботу.	Переваги у швидкості
2.	Спрощення інтерфейсу користувача	Простота роботи з системою.	Користувачам не потрібно читати документацію, інтерфейс є інтуїтивним.
3.	Точність результатів	Точні результати обробки даних.	Сервіс надає точніші результати прогнозування.

Розробимо трирівневу маркетингова модель товару, а саме уточнимо ідею продукту та/або послуги, його фізичні складові, особливості процесу його надання. Результати наведено у таблиці 4.19. В таблиці 4.19 описано три рівні моделі товару. Зазначено основний задум товару та перераховано основні властивості продукту та його характеристики, зазначено, що новим користувачам буде надано місячний період безкоштовно, а потенційний товар буде захищено від копіювання через використання веб-сервісного підходу та ліцензіювання. Тестування додатку буде проведено з використанням стандарту ISO 4444, маркування для товару відсутнє і виконується постійна підтримка користувачів, для початку користувачу надається місячна пробна версія.

Таблиця 4.19 – Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сукупність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Розробка програми-парсеру, що може виконувати прогнози на майбутні значення послідовностей, на основі даних з соціальних мереж як гетерогенних джерел даних.</td></tr>
<tr><td rowspan="5">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх/Тл/Е/Ор</td></tr>
<tr><td>1. Простота інтерфейсу користувача
2. Швидкість роботи
3. Безпека</td><td>Не матеріальна</td><td>Технологічна</td></tr>
<tr><td colspan="3">Якість: Згідно до стандарту ISO 4444 буде проведено тестування</td></tr>
<tr><td colspan="3">Маркування відсутнє.</td></tr>
<tr><td colspan="3">Моя компанія: “Web Apps”</td></tr>
<tr><td rowspan="2">III. Товар із підкріплюванням</td><td colspan="3">Місячна пробна версія</td></tr>
<tr><td colspan="3">Постійна підтримка для користувачів</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: використання веб-сервісного підходу, товар не встановлюється на ПК користувача, ліцензіювання.</td></tr>
</table>

Визначимо цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів. Результати наведено в таблиці 4.20.

Як видно з таблиці 4.20 даний продукт матиме значно нижчу ціну за підписку на рік ніж в конкурентів, що дасть перевагу при виході на ринок та надасть можливість для отримання більшості клієнтів.

Таблиця 4.20 Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на послугу
1.	1250$	1500$	10000$	1000$

Визначимо оптимальну систему збуту для даного сервісу, таблиця 4.21.

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Оплата щомісячної або річної підписки на використання ПЗ.	Продаж	Однорівневий	Власні сили та через посере-дників.

Як видно з таблиці 4.21 було вирішено проводити збут власними силами та залучати сторонніх посередників, використовуючи однорівневий канал збуту, а специфікою закупівельної поведінки було обрано використання щомісячної та річної підписки, що є основною для веб-сервісих додатків.

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів.

В таблиці 4.22 описано концепцію маркетингових комунікацій даного проекту. Описано специфіку поведінки цільових клієнтів, канали комунікації для отримання зворотного зв’язку. Описано ключові пункти, що характеризують проект. Наведено завдання рекламного повідомлення та концепцію рекламного

звернення. Було прийнято рішення, що для реклами буде використано демо-відео з використанням сервісу, в якому буде показано основні переваги сервісу.

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціювання	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Оформлення підписки на використання ПЗ в інтернеті	Інтернет	Швидкодія, простота використання, безпека, точність.	Показати переваги ПЗ, у тому числі перед конкурентами.	Демо-відео із використаням.

4.5 Висновки за розділом

В даному розділі було проведено аналіз програмного продукту у якості стартап проекту. Можна зазначити що у проекта є можливість комерціалізації, адже ринок надання послуг в мережі інтернет з використанням рекомендаційних систем динамічно розвивається, створюються нові додатки які, в свою чергу, стимулюють попит на різноманітні допоміжні засоби для пришвидшення роботи та оптимізації алгоритмів та матеріального забезпечення.

Було проведено аналіз ризиків та можливостей які можуть виникнути. Основними загрозами, очікувано, виявились конкуренція та зміна потреб користувачів. Найбільш вдалими можливостями для нас, звичайно ж, є невдачі наших конкурентів. Також гарною можливістю для росту є загальне «підняття» ринку.

На ринку наявна нецінова конкуренція, існує декілька фірм-конкурентів, але всі вони покривають лише якусь певну частину функціональності нашої системи, тому вихід на нього буде потребувати певних зусиль та

капіталовкладень. Проте проект є доволі конкурентноспроможним завдяки своїй нижчій собівартості та значно більшій кількості функціоналу. Через те, що він є повністю програмним, його розробка не потребує витрат на різноманітні матеріали та обладнання, необхідні для виготовлення корпусу, схем, тощо.

Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту за допомогою хмарного середовища Google App Engine та використанням Apache Spark для обробки великих масивів даних, а потім якісну рекламу та PR, сконцентровану навколо позитивних характеристиках даного програмного продукту, таких як низька ціна, більш істотний ефект покращення якості видачі, кросплатформеність і т.д.

З огляду на проведений аналіз, можна чітко сказати, що подальша імплементація проекту є доцільною, адже він може знайти свою цільову аудиторію та зайняти місце на ринку.

ВИСНОВОК

В даній дипломній роботі було розглянуто основні принципи та характеристики гетерогенних джерел даних на прикладі соціальних мереж, в контексті даної роботи було розглянуто соціальні мережі з пошуку роботи та викладення вакансій з метою отримання статистики та даних сучасного ринку інформаційних технологій, а саме популярність спеціальностей розробників, використання фреймворків та технологій та їх популярність, для виконання цього завдання було використано найпопулярнішу мережу для пошуку роботи – LinkedIn, а також українську мережу Dou, та популярні сервіси Stackoverflow Jobs та Jooble.

Показано, що розробка додатків для таких систем не є простою задачею, це пов’язано із даними таких систем, що постійно оновлюються в режимі реального часу, а їх кількість є надвеликою, для реалізації та обробки даних з таких систем потрібно використовувати відповідні інструменти та архітектуру.

Було продемонстровано основні алгоритми для прогнозування послідовності даних, а саме ARIMA, лінійну регресію та її вдосконалення, рекурсивні нейронні мережі, а також метод експоненційного спуску, що хоч є простим в реалізації, проте його вдосконалення дають ефективні результати. Хоча кожен з цих методів має свої переваги та недоліки, на реальному додатку було продемонстровано їх роботу та досліджено їх основні принципи роботи.

Для обробки великих масивів даних було обрано фреймворк Spark, що здатен обробляти величезні обсяги даних на кластерах обчислювальних машин, як було показано у другій частині диплому, даний вибір мав найбільше переваг для використання у практичній частині, обійшовши Apache MapReduce, MPI та можливість створення власного аналогу з використанням декількох серверів з SQL базами даних, такий підхід було відкинуто через великий обсяг часу, що потрібно буде затратити на розробку та використання SQL баз даних.

При плануванні розробки додатку було вирішено розробити веб-сервісний додаток, основними перевагами якого є те, що користувачам не потрібно буде розгортати та встановлювати додаток на власних обчислювальних машинах, не використовуючи обчислювальні ресурси користувача. Було продемонстровано ряд альтернатив для використання архітектури такого веб-додатку, а саме монолітна та мікросервісна, проте було обрано мікросервісний підхід, який, хоч і має ряд недоліків, однак в даному випадку добре підходить, а можливість розбиття додатку на маленькі компоненти з майбутнім розширенням його компонентів є чудовою альтернативою в даному випадку.

В третій частині диплому було продемонстровано основні компоненти розробленого додатку. Розроблена програма-парсер складається з шести сервісів, що мають свою функціональність та розділені між собою. Сервіс прогнозування використовує описані підходи для прогнозування даних послідовностей в майбутньому, а сервіс отримання даних з соціальних мереж, використовуючи паралельну обробку даних, отримує та парсить дані з соціальних мереж, в результаті чого передає його на відповідний сервіс обробки даних, де, в залежності від розміру даних та складності задачі дані можуть бути оброблені одразу чи передаються на Spark. Сервіс взаємодії користувача з системою є чи не найважливішим з точки зору користувача і з’єднує усі сервіси у єдину екосистему так, що користувач вважає, що працює з одним додатком, а не цілим рядом виокремлених підсистем-сервісів.

В четвертій частині диплому було розглянуто розроблену систему з точки зору ринкових можливостей, економічної привабливості. Було розроблено маркетингову програму та проаналізовано ризики, розраховано витриати. В результаті чого було встановлено, що розробка такого додатку є економічно вигідною.

ПЕРЕЛІК ПОСИЛАНЬ

1. Social media definition and the governance challenge: An introduction to the special issue [Text] / Obar, Jonathan A.; Wildman, Steve – Telecommunications policy, 2005. 745–750 p.;
2. History and Different Types of Social Media [Electronic resource] – University of Southem California, 2018 – Режим доступу: http://scalar.usc.edu/works/everything-you-always-wanted-to-know-about-social-media-but-were-too-afraid-to-ask/history-and-different-types-of-social-media – Назва з екрану;
3. Феномен соціальної мережі в інформаційному середовищі [Електронний ресурс] / А. Кромська // Науковий блог Національного університету “Острозька академія”, 2015 – Режим доступу: https://naub.oa.edu.ua/2015/феномен-соціальної-мережі-в-інформац – Назва з екрану;
4. Палій С.В. Соціальні мережі як засіб комунікації електронного навчання. / С. В. Палій // Управління розвитком складних систем. – 2013. – Вип. 13. – с. 152–156;
5. Шапіро О.О. Масова комунікація в on-line вимірі: зміна парадигми [Електронний ресурс] / О. О. Шапіро // Вісник Національної юридичної академії України імені Ярослава Мудрого, 2013. – № 2. – с. 57–65. – Режим доступу: http://nbuv.gov.ua/j-pdf/Vnyua_2013_2_9.pdf;
6. A Survey Of Data Mining Techniques for Social Network Analysis [Electronic resource] / Adedoyin-Olowe M, Gaber M., Stahl F. – School of Computing Science and Digital Media, Robert Gordon University – Режим доступу: https://jdmdh.episciences.org/18/pdf;
7. Sentiment Analysis in Social Networks: A Machine Learning Perspective [Electronic Resource] / E. Fersini – ScienceDirect – Режим доступу: https://www.sciencedirect.com/science/article/pii/B9780128044124000061b – Назва з екрану;

8. Machine Learning for Social Network Analysis: A Systematic Literature Review [Electronic Resource] / Sagar S. De, Gi-Nam Wang – Research Gate – Режим доступу: https://www.researchgate.net/publication/251236864_Machine_Learning_for_Social_Network_Analysis_A_Systematic_Literature_Review – Назва з екрану;
9. A Methodology for Temporal Analysis of Evolving Concepts in Twitter. [Text] / Adedoyin-Olowe, M., Gaber, M., Stahl, F.: // Proceedings of the 2013 ICAISC, International Conference on Artificial Intelligence and Soft Computing, 2013.
10. Simple Exponential Smoothing [Electronic resource] – OTexts, Режим доступу – https://www.otexts.org/fpp/7/1 – Назва з екрану;
11. Exponential Smoothing for Predicting Demand [Text] / Brown, Robert G. // Cambridge, Massachusetts – 1956, p. 15;
12. Forecasting Trends and Seasonal by Exponentially Weighted Averages [Text] / Holt, Charles C. // Office of Naval Research Memorandum – 1957, 5–10 p.;
13. Introduction to Linear Regression [Electronic resource] / David M. Lane // OnlineStatBook – Режим доступу: http://onlinestatbook.com/2/regression/intro.html – Назва з екрану;
14. Neural Networks for Time Series Prediction [Electronic resource] / Touretzky D., Laskowski K. // – Artigical Neural Networks , 2006 – Режим доступу: https://www.cs.cmu.edu/afs/cs/academic/class/15782-f06/slides/timeseries.pdf;
15. Introduction to ARIMA: nonseasonal models [Electronic resource] – DukePeople – Режим доступу: https://people.duke.edu/~rnau/411arim.htm – Назва з екрану;
16. What is Apache Spark? The big data analytics plarform explained [Electronic resource] / Pointer I. // InfoWorld – 2017 – Режим доступу: https://www.infoworld.com/article/3236869/analytics/what-is-apache-spark-the-big-data-analytics-platform-explained.html – Назва з екрану;
17. What is MapReduce? [Electronic resource] – IBM Analytics – Режим доступу: https://www.ibm.com/analytics/hadoop/mapreduce – Назва з екрану;
18. Клієнт-серверна архітектура та ролі серверів [Електронний ресурс] / Змерзлий І. // Medium – 2017 – Режим доступу:

https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229 – Назва з екрану;

19. Переваги мікросервісів та їх створення за допомогою .NET [Електронний ресурс] // InternetDevels – 2016 – Режим доступу: https://internetdevels.ua/blog/building-microservices-dotnet – Назва з екрану.
20. Сергеєв Є.І. Дослідження гетерогенних джеред даних на прикладі соціальних мереж та реалізація їх обробки і прогнозування даних [Електронй ресурс] // Міжнародний науковий журнал “Інтернаука”. – 2018. – №8 – Режим доступу: https://www.inter-nauka.com/uploads/public/15247731129737.pdf.